





First Edition (September 1989)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used.

Requests for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

© Copyright International Business Machines Corporation 1986, 1989. All Rights Reserved.

Note to US Government users — Documentation related to Restricted Rights — Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Special Notices

The following names, used in this publication, are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries:

IBM
Operating System/2
OS/2
Presentation Manager
Systems Application Architecture.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license enquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Armonk, NY 10504.

Library Structure

OS/2 Product

IBM Operating System/2
Standard Edition
Version 1.2

Getting Started
Using Advanced Features
Product Information

6024926 3.5" diskette
6024930 5.25" diskette

OS/2 Programming Tools and Information

IBM Operating System/2
Version 1.2

Installation
Programming Overview
Programming Guide
Building Programs
Dialog Tag Language
Guide and Reference
Dialog Manager Guide
and Reference
Dialog Manager and
Dialog Tag Language
Reference Summary
Programming Reference
(3 books)
Bindings Reference for
Presentation Manager
(4 books for COBOL/2,
FORTRAN/2, C/2, and
Macro Assembler/2)
I/O Subsystems and
Device Support
(2 books)
Systems Application
Architecture
Common User Access:
Advanced Interface
Design Guide

6024929

Separate Order (no charge)

Keyboards and Code Pages

6280345

Available Separately

Command Reference

6024928

Service Coordinator's
Guide

15F2214

Programming Languages

IBM Basic Compiler/2
Version 1.0 6280179

IBM Macro Assembler/2
Version 1.0 6280181

IBM Pascal Compiler/2
Version 1.0 6280183

IBM FORTRAN/2
Version 1.02 6280185

IBM COBOL/2
Version 1.0 6280207

IBM C/2
Version 1.1 6280284

Systems Application Architecture

An Overview

GC26-4341

Writing Applications:
A Design Guide

SC26-4362

Common User Access:
Advanced Interface
Design Guide
SC26-4582

Common User Access:
Basic Interface Design
Guide
SC26-4583

Common Programming Interface

C Reference - Level 2

SC09-1308

COBOL Reference

SC26-4354

Dialog Reference

SC26-4356

FORTRAN Reference

SC26-4357

Procedures Language
Reference
SC26-4358

Presentation Reference

SC26-4359

Contents

Chapter 1. Introduction	1-1
Header Files	1-1
Dynamic Linking	1-1
IBM Operating System/2 Call Format and Characteristics	1-2
Standard Data Types Used in this Document	1-3
Mapping Metalanguage Data Types to MASM Data Types	1-3
Addressing Elements in Arrays	1-3
 Chapter 2. Data Types	 2-1
 Chapter 3. Device Function Calls	 3-1
 Chapter 4. Graphics Presentation Interface Function Calls	 4-1
 Chapter 5. Picture Function Calls	 5-1
 Chapter 6. Profile Function Calls	 6-1
 Chapter 7. Spooler Function Calls	 7-1
 Chapter 8. Video Function Calls	 8-1
 Chapter 9. Window Function Calls	 9-1
 Chapter 10. Functions Supplied by Applications	 10-1
 Index	 X-1

Chapter 1. Introduction

This book describes the Operating System/2 (OS/2) Version 1.2 language binding for IBM Macro Assembler/2 (MASM).

It provides language-dependent information about the functions described in the OS/2 Version 1.2 *Presentation Manager Programming Reference*.

The information in this book, together with the functional descriptions in the *Presentation Manager Programming Reference*, enable the user to generate call statements in MASM.

The following information is provided:

- The syntax for each data type and structure
- The parameter list for each call.

Header Files

All functions require an "include" for the system header file OS2.INC:

```
include OS2.INC
```

Also, most functions require a "define" to select an appropriate (conditional) section of the header file, and hence, the required entry point. Where this is necessary, it is shown at the head of the function definition in the form:

```
INCL_name equ 1
```

Programming Note: These "defines" must precede the "include OS2.INC."

Two equates occur in the header files for each constant that represents a 32-bit quantity. These are "xxxx_HI" which is the high-order word of the constant, and "xxxx_LO" which is the low-order word, (where "xxxx" is the name of the constant given in the *Programming Reference*).

Dynamic Linking

The processor's protect-mode call architecture gives various advantages over the static-module structure of DOS. In particular, these are:

- Application programs need only load the most commonly used segments when they are started. Exception-processing routines do not need to be loaded as they can be called (and loaded automatically) as necessary.
- Dynamic-link package updates can be transparent to their clients.

The actual programming steps required to use the dynamic link feature are as follows:

1. The programmer codes a call to a subroutine that is to be dynamically linked and declares it "EXTERN name:FAR."
2. The compiler generates a standard external reference.
3. When the object module is linked, the linker is provided with the names of libraries containing dynamic link definition records. These records provide correspondence between the called entry point and the module file containing the routine being called.

IBM Operating System/2 Function Calls

IBM Operating System/2 applications must use the dynamic link mechanism (FAR CALL) to get to all services.

IBM Operating System/2 Function Call Rules

Rules for the IBM Operating System/2 interface are shown below:

- Rule 1:** All parameters are passed on the stack (SS:SP).
- Remarks** Passing parameters on the stack is consistent across a broad base of languages on the 80286/80386 family of processors. The minimum recommended available stack space is 2 KB.
- Rule 2:** All interfaces pass a return code back to the caller in AX if the returned value is a WORD, or in DX:AX if the returned value is a DWORD.
- Remarks** The use of register AX as a call return code is also consistent across many languages. For calls prefixed by DOS, KBD, MOU, and VIO (see *Control Program Programming Reference*) all other user registers except the FLAGS register are preserved. The contents of the FLAGS register are undefined. The state of the direction flag in the FLAGS register is preserved, or if the direction flag is clear when an API is called, it is clear when the API returns.
- Rule 3:** All addresses of **output** parameters are of the form:
- selector:offset.
- Remarks** Fully-qualified addresses are available across all memory models as a method for returning values to a requestor. This allows one function entry point to serve all languages and memory models.
- Rule 4:** Each function is accessed by a FAR CALL.
- Remarks** This is a requirement for the calls to be dynamic link entries.
- Rule 5:** All calls remove the parameters from the stack.
- Remarks** Parameter lists are fixed length on a function basis. Variable-length parameter lists are not supported.
- Rule 6:** All call names must be uppercase when linked.
- Remarks** If a compiler or assembler generates case-sensitive (uppercase and lowercase) external references, all calls or call definitions must be in **uppercase** characters.

IBM Operating System/2 Call Format and Characteristics

The call descriptions follow a pseudo assembly language format. The interfaces are shown to be descriptive rather than to be an example of a coding sequence. The conventions described here are used throughout.

Call Format

Since all parameters are pushed onto the stack, there are several pseudo operations to describe these operations:

- **PUSH** — push an item onto the stack.
This can be used to push various size items onto the stack.
- **PUSH@** — push the address of an item onto the stack.
All addresses in these interfaces are composed of a 32-bit value: a 16-bit selector and a 16-bit offset. This address can point to any of the data item types.
- **CALL** — call a function.
All function calls are accessed via FAR CALLs. This is a requirement when using the Dynamic Link mechanism.

Programming Note: Note that the format of the function calls is illustrative only. For example, there is no actual 'PUSH@' instruction in IBM Macro Assembler/2, and 'PUSH DWORD' implies that two words must be pushed, the high-order word first.

Standard Data Types Used in this Document

Below is a list of the standard data types used in this document to make up a parameter list:

Type	Description
WORD	2 bytes. This type of operand can be passed by value (pushed onto the stack) or by reference (the address of the operand is pushed onto the stack).
DWORD	4 bytes. This type of operand can be passed by value (pushed onto the stack as two sequential 16-bit values) or by reference (the address of the operand is pushed onto the stack).
ASCIIZ	Null (0) terminated character string. This type of operand can only be accessed by reference.
Other	Any other structure. This type of operand can only be accessed by reference.

Mapping Metalanguage Data Types to MASM Data Types

Some data types shown in Chapter 2 of this book do not occur in MASM, but belong to the metalanguage data types described in Chapter 2 of the *Presentation Manager Programming Reference*. They are included to provide a mapping to the corresponding MASM data types, and can be identified as those that contain the phrase "(Maps via...)."

This treatment is not needed for every metalanguage data type, as many of them have the same name as the corresponding MASM data type.

Addressing Elements in Arrays

Constants defining array elements are given values that are zero-based in MASM; that is, the numbering of the array elements starts at zero, not one.

For example, in the DevQueryCaps function described in Chapter 3 of the *Presentation Manager Programming Reference*, the sixth element of the **Array** parameter is CAPS_HEIGHT, which is equated to 5.

Count parameters related to such arrays always mean the actual number of elements available. Therefore, again using the DevQueryCaps function as an example, if all elements up to and including CAPS_HEIGHT are provided for, **Count** could be set to (CAPS_HEIGHT + 1).

In functions for which the starting array element can be specified, this is always zero-based, and the MASM element number constants can be used directly. For example, to start with the CAPS_HEIGHT element, the **Start** parameter can be set to CAPS_HEIGHT.

Chapter 2. Data Types

ACCEL

Accelerator structure.

```
ACCEL struc
    acc_fs      dw ?    ;Options
    acc_key     dw ?    ;Key
    acc_cmd     dw ?    ;Command code
ACCEL ends
```

ACCELTABLE

Accelerator-table structure.

```
ACCELTABLE struc
    acct_cAccel dw ?    ;Number of accelerator entries
    acct_codepage dw ?  ;Code page for accelerator entries
    acct_aaccel dw (size ACCEL)/2 * ACCT_CACCEL dup (?)
                                ;Accelerator entries
ACCELTABLE ends
```

ARCPARAM

Arc-parameters structure.
(Maps via ARCPARAMS)

ARCPARAMS

Arc-parameters structure.

```
ARCPARAMS struc
    arcp_lP     dd ?    ;P coefficient
    arcp_lQ     dd ?    ;Q coefficient
    arcp_lR     dd ?    ;R coefficient
    arcp_lS     dd ?    ;S coefficient
ARCPARAMS ends
```

AREABUNDLE

Area-attributes bundle structure.

```
AREABUNDLE struc
    pbnd_lColor      dd ?    ;Area foreground color
    pbnd_lBackColor  dd ?    ;Area background color
    pbnd_usMixMode    dw ?    ;Area foreground-mix mode
    pbnd_usBackMixMode dw ?    ;Area background-mix mode
    pbnd_usSet        dw ?    ;Pattern set
    pbnd_usSymbol     dw ?    ;Pattern symbol
    pbnd_ptlRefPoint  dw (size POINTL)/2 dup (?)
                                ;Pattern reference point
AREABUNDLE ends
```

ATOM

Atom identity.
(Scalar type: Maps to word)

BANDRECT

Rectangle structure, used for the coordinates of an output band (see DevEscape).

An empty rectangle is one for which **xleft** is greater than **xright**, or **ybottom** is greater than **ytop**.

```
BANDRECT struc
    lxFleft dd ?    ;x coordinate of left edge of rectangle
    lyBottom dd ?    ;y coordinate of bottom edge of rectangle
    lxFright dd ?    ;x coordinate of right edge of rectangle
    lyTop dd ?    ;y coordinate of top edge of rectangle
BANDRECT ends
```

BITMAPINFO

Bit-map information structure.

Each bit plane logically contains (**bitmapwidth*bitmapheight*bitcount**) bits, although the actual length can be greater because of padding.

BITMAPINFO struc

```

bmi_cbFix      dd  ?    ;Length of fixed portion of structure
bmi_cx         dw  ?    ;Bit-map width in pels
bmi_cy         dw  ?    ;Bit-map height in pels
bmi_cPlanes    dw  ?    ;Number of bit planes
bmi_cBitCount  dw  ?    ;Number of bits per pel within a plane
bmi_argbColor  dw (size RGB)/2 dup (?) ;Array of RGB values

```

BITMAPINFO ends

BITMAPINFOHEADER

Bit-map information header structure.

Each bit plane logically contains (**bitmapwidth*bitmapheight*bitcount**) bits, although the actual length can be greater because of padding.

BITMAPINFOHEADER struc

```

bmp_cbFix      dd  ?    ;Length of structure
bmp_cx         dw  ?    ;Bit-map width in pels
bmp_cy         dw  ?    ;Bit-map height in pels
bmp_cPlanes    dw  ?    ;Number of bit planes
bmp_cBitCount  dw  ?    ;Number of bits per pel within a plane

```

BITMAPINFOHEADER ends

BIT8

Defines eight independent BOOL values.
(Maps via UCHAR)

BIT16

Defines 16 independent BOOL values.
(Maps via USHORT)

BIT32

Defines 32 independent BOOL values.
(Maps via ULONG)

BOOL

Boolean.

The *BOOL* data type is defined such that a nonzero value indicates TRUE, and a zero value indicates FALSE.

The constants TRUE and FALSE have been defined as 1 and 0 respectively. The constant TRUE should never be used in an equality comparison with a *BOOL* as this may result in incorrect results.

When an application needs to pass a TRUE value on a function call, or as a return value from a window procedure, it is recommended that it passes the value TRUE, which is 1, and not a nonzero value.

(Scalar type: Maps to word)

BTNCDATA

Button-control-data structure.

BTNCDATA struc

```

btncd_cb          dw  ?    ;Length of control data in bytes
btncd_fsCheckState dw  ?    ;Check state of button
btncd_fsHiliteState dw  ?    ;Highlighting state of button

```

BTNCDATA ends

BUFFER

Buffer.

(Maps via PBYTE)

BUNDLE	<p>Bundle data area. It is overlaid by one of these attribute bundle structures:</p> <p><i>AREABUNDLE</i> <i>CHARBUNDLE</i> <i>IMAGEBUNDLE</i> <i>LINEBUNDLE</i> <i>MARKERBUNDLE</i>.</p> <p>(Maps via PBUNDLE)</p>
BYTE	<p>Byte.</p> <p>(Scalar type: Maps to byte)</p>
CATCHBUF	<p>Saved execution environment buffer.</p> <pre>CATCHBUF struc ctchbf_reserved dd 4 dup (?) ;Save area CATCHBUF ends</pre>
CHAR	<p>Single-byte character.</p> <p>(Scalar type: Maps to byte)</p>
CHARBUNDLE	<p>Character-attributes bundle structure.</p> <pre>CHARBUNDLE struc cbnd_lColor dd ? ;Character foreground color cbnd_lBackColor dd ? ;Character background color cbnd_usMixMode dw ? ;Character foreground-mix mode cbnd_usBackMixMode dw ? ;Character background-mix mode cbnd_usSet dw ? ;Character set cbnd_usPrecision dw ? ;Character precision cbnd_sizfxCell dw (size SIZEF)/2 dup (?) ;Character cell size cbnd_ptlAngle dw (size POINTL)/2 dup (?) ;Character angle cbnd_ptlShear dw (size POINTL)/2 dup (?) ;Character shear cbnd_usDirection dw ? ;Character direction CHARBUNDLE ends</pre>
CLASSINFO	<p>Class-information structure.</p> <pre>CLASSINFO struc clsi_fClassStyle dd ? ;Class-style flags clsi_pfnWindowProc dd ? ;Window procedure clsi_cbWindowData dw ? ;Number of additional window words CLASSINFO ends</pre>
COUNT2	<p>Count in the range 0 through 65 535.</p> <p>(Maps via USHORT)</p>
COUNT2B	<p>Count of bytes, in the range 0 through 65 535.</p> <p>(Maps via USHORT)</p>
COUNT2CH	<p>Count of characters, in the range 0 through 65 535.</p> <p>(Maps via USHORT)</p>
COUNT4	<p>Count in the range 0 through 4 294 967 295.</p> <p>(Maps via ULONG)</p>
COUNT4B	<p>Count of bytes, in the range 0 through 4 294 967 295.</p> <p>(Maps via ULONG)</p>
CPID	<p>Code-page identity.</p> <p>(Maps via USHORT)</p>
CREATEPARAMS	<p>Create parameters structure.</p> <p>(Maps via STORAGE)</p>

CREATESTRUCT

Create-window data structure.

```

CREATESTRUCT struc
    crst_pPresParams      dd ?      ;Presentation parameters
    crst_pCtlData         dd ?      ;Control data
    crst_id               dw ?      ;Window identifier
    crst_hwndInsertBehind dd ?      ;Window behind which the window
                                is to be placed
    crst_hwndOwner        dd ?      ;Window owner
    crst_cy               dw ?      ;Window height
    crst_cx               dw ?      ;Window width
    crst_y               dw ?      ;y coordinate of origin
    crst_x               dw ?      ;x coordinate of origin
    crst_fStyle           dd ?      ;Window style
    crst_pszText          dd ?      ;Window text
    crst_pszClass         dd ?      ;Registered window class name
    crst_hwndParent       dd ?      ;Parent window handle
CREATESTRUCT ends

```

CTLDATA

Class-specific control data, beginning with a value conforming to a *COUNT2B* data type, which specifies the overall length of the data.

The following are cases of this data type:

```

BTNCDATA      Button control data.
ENTRYFDATA    Entry field control data.
FRAMECDATA    Frame control data.
SBCDATA       Scroll bar control data.
MLECDATA      Multi-line entry field control data.
(Maps via PVOID)

```

CURSORINFO

Cursor-information structure.

```

CURSORINFO struc
    csri_hwnd dd ?      ;Window handle
    csri_x    dw ?      ;x coordinate
    csri_y    dw ?      ;y coordinate
    csri_cx   dw ?      ;Cursor width
    csri_cy   dw ?      ;Cursor height
    csri_fs   dw ?      ;Options
    csri_rcClip dw (size RECTL)/2 dup (?) ;Cursor box
CURSORINFO ends

```

DDEINIT

Dynamic-data-exchange initiation structure.

```

DDEINIT struc
    ddei_cb      dw ?      ;Length of structure
    ddei_pszAppName dd ?    ;Application name
    ddei_pszTopic dd ?      ;Topic
DDEINIT ends

```

DDESTRUCT

Dynamic-data-exchange control structure.

```

DDESTRUCT struc
    dde_cbData      dd ?      ;Total length
    dde_fsStatus     dw ?      ;Status
    dde_usFormat     dw ?      ;Data format
    dde_offszItemName dw ?      ;Offset to item
    dde_offabData     dw ?      ;Offset to beginning of data
DDESTRUCT ends

```

DEVOPENDATA

Open-device data area. The format of this area is the same as a *DEVOPENSTRUC* structure.
(Maps via *PDEVOPENDATA*)

DEVOPENSTRUC

Open-device data structure.

The same structure is applicable to both DevOpenDC and SpiQmOpen calls.

```

DEVOPENSTRUC struc
    dop_pszLogAddress      dd ?    ;Logical address
    dop_pszDriverName      dd ?    ;Driver name
    dop_pdriv              dd ?    ;Driver data
    dop_pszDataType        dd ?    ;Data type
    dop_pszComment         dd ?    ;Comment
    dop_pszQueueProcName   dd ?    ;Queue-processor name
    dop_pszQueueProcParams dd ?    ;Queue-processor parameters
    dop_pszSpoolerParams   dd ?    ;Spooler parameters
    dop_pszNetworkParams   dd ?    ;Network parameters
DEVOPENSTRUC ends

```

DLGTEMPLATE

Dialog-template structure.

```

DLGTEMPLATE struc
    dlgt_cbTemplate        dw ?    ;Length of template
    dlgt_type              dw ?    ;Template format type
    dlgt_codepage          dw ?    ;Code page
    dlgt_offadlgti         dw ?    ;Offset to dialog items
    dlgt_fsTemplateStatus  dw ?    ;Template status
    dlgt_iItemFocus        dw ?    ;Index of item to receive focus
                                initially
    dlgt_coffPresParams    dw ?    ;Count of presentation-parameter
                                offsets
    dlgt_adlgti            dw (size DLGTITEM)/2 dup (?) ;Start of
                                dialog items
DLGTEMPLATE ends

```

DLGTITEM

Dialog-item structure.

```

DLGTITEM struc
    dlgti_fsItemStatus     dw ?    ;Status
    dlgti_cChildren        dw ?    ;Count of children to this dialog
                                item
    dlgti_cchClassName     dw ?    ;Length of class name
    dlgti_offClassName     dw ?    ;Offset to class name
    dlgti_cchText          dw ?    ;Length of text
    dlgti_offText          dw ?    ;Offset to text
    dlgti_flStyle          dd ?    ;Dialog item window style
    dlgti_x                dw ?    ;x coordinate of origin of
                                dialog-item window
    dlgti_y                dw ?    ;y coordinate of origin of
                                dialog-item window
    dlgti_cx               dw ?    ;Dialog-item window width
    dlgti_cy               dw ?    ;Dialog-item window height
    dlgti_id               dw ?    ;Identity
    dlgti_offPresParams    dw ?    ;Reserved
    dlgti_offCtlData       dw ?    ;Offset to control data
DLGTITEM ends

```

DRIVDATA

Driver-data structure.

```

DRIVDATA struc
    driv_cb                dd ?    ;Length
    driv_lVersion          dd ?    ;Version
    driv_szDeviceName      db 32 dup (?) ;Device name
    driv_abGeneralData     db ?    ;General data
DRIVDATA ends

```

ENTRYFDATA

Entry field control data structure.

```
ENTRYFDATA struc
    efd_cb          dw  ?    ;Length of control data in bytes
    efd_cchEditLimit dw  ?    ;Edit limit
    efd_ichMinSel    dw  ?    ;Minimum selection
    efd_ichMaxSel    dw  ?    ;Maximum selection
ENTRYFDATA ends
```

ERRINFO

Error-information structure.

```
ERRINFO struc
    erri_cbFixedErrInfo dw  ?    ;Length of fixed data to this
                                structure
    erri_idError         dd  ?    ;Error identity
    erri_cDetailLevel    dw  ?    ;Number of levels of detail
    erri_offaoffszMsg     dw  ?    ;Offset to the array of message
                                offsets
    erri_offBinaryData   dw  ?    ;Offset to the binary data
ERRINFO ends
```

ERRORID

Error identity.

(Scalar type: Maps to **dword**)**FATTRS**

Font-attributes structure.

```
FATTRS struc
    fat_usRecordLength dw  ?    ;Length of record
    fat_fsSelection     dw  ?    ;Selection indicators
    fat_lMatch          dd  ?    ;Matched-font identity
    fat_szFacename      db  FACESIZE dup (?) ;Typeface name
    fat_idRegistry      dw  ?    ;Registry identifier
    fat_usCodePage      dw  ?    ;Code page
    fat_lMaxBaselineExt dd  ?    ;Maximum base-line extension
    fat_lAveCharWidth   dd  ?    ;Average character width
    fat_fsType          dw  ?    ;Type indicators
    fat_fsFontUse       dw  ?    ;Font-use indicators
FATTRS ends
```

FFDESCS

Font-file descriptor.

(Scalar type: Maps to **byte * FACESIZE * 2**)**FIXED**Signed-integer fraction (16:16). This can be treated as a *LONG* where the value has been multiplied by 65 536.(Scalar type: Maps to **dword**)

FONTMETRICS

Font-metrics structure.

```
FONTMETRICS struc
    fm_szFamilyname      db FACESIZE dup (?) ;Family name
    fm_szFacename        db FACESIZE dup (?) ;Facename
    fm_idRegistry         dw ? ;Registry identifier
    fm_usCodePage         dw ? ;Code page
    fm_lEmHeight          dd ? ;Em height
    fm_lXHeight           dd ? ;x height
    fm_lMaxAscender       dd ? ;Maximum ascender
    fm_lMaxDescender      dd ? ;Maximum descender
    fm_lLowerCaseAscent   dd ? ;Lowercase ascent
    fm_lLowerCaseDescent  dd ? ;Lowercase descent
    fm_lInternalLeading    dd ? ;Internal leading
    fm_lExternalLeading    dd ? ;External leading
    fm_lAveCharWidth      dd ? ;Average character width
    fm_lMaxCharInc        dd ? ;Maximum character increment
    fm_lEmInc             dd ? ;Em increment
    fm_lMaxBaselineExt    dd ? ;Maximum baseline extent
    fm_sCharSlope         dw ? ;Character slope
    fm_sInlineDir         dw ? ;Inline direction
    fm_sCharRot           dw ? ;Character rotation
    fm_usWeightClass      dw ? ;Weight class
    fm_usWidthClass       dw ? ;Width class
    fm_sXDeviceRes        dw ? ;x device resolution
    fm_sYDeviceRes        dw ? ;y device resolution
    fm_sFirstChar         dw ? ;First character
    fm_sLastChar          dw ? ;Last character
    fm_sDefaultChar       dw ? ;Default character
    fm_sBreakChar         dw ? ;Break character
    fm_sNominalPointSize  dw ? ;Nominal point size
    fm_sMinimumPointSize  dw ? ;Minimum point size
    fm_sMaximumPointSize  dw ? ;Maximum point size
    fm_fsType             dw ? ;Type indicators
    fm_fsDefn             dw ? ;Definition indicators
    fm_fsSelection        dw ? ;Selection indicators
    fm_fsCapabilities     dw ? ;Capabilities
    fm_lSubscriptXSize    dd ? ;Subscript x size
    fm_lSubscriptYSize    dd ? ;Subscript y size
    fm_lSubscriptXOffset  dd ? ;Subscript x offset
    fm_lSubscriptYOffset  dd ? ;Subscript y offset
    fm_lSuperscriptXSize  dd ? ;Superscript x size
    fm_lSuperscriptYSize  dd ? ;Superscript y size
    fm_lSuperscriptXOffset dd ? ;Superscript x offset
    fm_lSuperscriptYOffset dd ? ;Superscript y offset
    fm_lUnderscoreSize    dd ? ;Underscore size
    fm_lUnderscorePosition dd ? ;Underscore position
    fm_lStrikeoutSize     dd ? ;Strikeout size
    fm_lStrikeoutPosition dd ? ;Strikeout position
    fm_sKerningPairs      dw ? ;Kerning pairs
    fm_sFamilyClass       dw ? ;Family class
    fm_lMatch             dd ? ;Matched font identity
FONTMETRICS ends
```

FRAMECDATA	<p>Frame-control-data structure.</p> <pre> FRAMECDATA struc fcdata_cb dw ? ;Length fcdata_fICreateFlags dd ? ;Frame-creation flags fcdata_hmodResources dw ? ;Identifier of required resource fcdata_idResources dw ? ;Resource identifier FRAMECDATA ends </pre>
GRADIENT	<p>Direction-vector structure. (Maps via GRADIENTL)</p>
GRADIENTL	<p>Direction-vector structure.</p> <pre> GRADIENTL struc gradl_x dd ? ;x component of direction gradl_y dd ? ;y component of direction GRADIENTL ends </pre>
HAB	<p>Anchor-block handle. (Maps via HANDLE)</p>
HACCEL	<p>Accelerator-table handle. (Maps via HANDLE)</p>
HANDLE	<p>The handle of a resource. (Maps via LHANDLE)</p>
HAPP	<p>Handle of an application started by the WinInstStartApp function. (Maps via HANDLE)</p>
HATOMTBL	<p>Atom-table handle. (Maps via HANDLE)</p>
HBITMAP	<p>Bit-map handle. (Maps via HANDLE)</p>
HCINFO	<p>Hardcopy-capabilities structure.</p> <pre> HCINFO struc hci_szFormname db 32 dup (?) ;Form name hci_cx dd ? ;Width (left-to-right) in millimeters hci_cy dd ? ;Height (top-to-bottom) in millimeters hci_xLeftClip dd ? ;Left clip limit in millimeters hci_yBottomClip dd ? ;Bottom clip limit in millimeters hci_xRightClip dd ? ;Right clip limit in millimeters hci_yTopClip dd ? ;Top clip limit in millimeters hci_xPels dd ? ;Number of pels between left and right clip limits hci_yPels dd ? ;Number of pels between bottom and top clip limits hci_fIAttributes dd ? ;Attributes of the form identifier HCINFO ends </pre>
HDC	<p>Device-context handle. (Maps via HANDLE)</p>
HENUM	<p>Window-enumeration handle. (Maps via HANDLE)</p>
HHEAP	<p>Heap handle. (Maps via HANDLE)</p>
HINI	<p>Initialization-file handle. (Maps via HANDLE)</p>
HLIB	<p>Library handle. (Maps via HANDLE)</p>
HMF	<p>Metafile handle. (Maps via HANDLE)</p>

HMODULE	Module handle. (Scalar type: Maps to word)
HMQ	Message-queue handle. (Maps via HANDLE)
HPOINTER	Handle of a pointer. (Maps via HANDLE)
HPROC	Processor handle. (Maps via HANDLE)
HPROGARRAY	Array of program handles. <pre> HPROGARRAY struc hpga_ahprog dd ? ;Program handle array HPROGARRAY ends </pre>
HPROGRAM	Program handle. (Maps via HANDLE)
HPS	Presentation-space handle. (Maps via HANDLE)
HRGN	Region handle. (Maps via HANDLE)
HSEM	Semaphore handle. (Scalar type: Maps to dword)
HSPL	Spooler handle. (Maps via HANDLE)
HSWITCH	Switch-list-entry handle. (Maps via HANDLE)
HVPS	VIO presentation-space handle. (Scalar type: Maps to word)
HWND	Window handle. (Maps via HANDLE)
IDENTITY	Identity value. Up to 65 536 different identities are available. (Maps via USHORT)
IDENTITY4	Identity value. Up to 4 294 967 296 different identities are available. (Maps via ULONG)
IMAGEBUNDLE	Image-attributes bundle structure. <pre> IMAGEBUNDLE struc ibmd_lColor dd ? ;Image foreground color ibmd_lBackColor dd ? ;Image background color ibmd_usMixMode dw ? ;Image foreground-mix mode ibmd_usBackMixMode dw ? ;Image background-mix mode IMAGEBUNDLE ends </pre>
INDEX2	Index value, in the range 0 through 65 535. (Maps via USHORT)
IPT	Insertion point for multi-line entry field. (Maps via LONG)
KERNINGPAIRS	Kerning-pair records structure. <pre> KERNINGPAIRS struc krnpr_sFirstChar dw ? ;First character of pair krnpr_sSecondChar dw ? ;Second character of pair krnpr_sKerningAmount dw ? ;Amount of kerning for this pair KERNINGPAIRS ends </pre>
LENGTH2	Length value, in the range 0 through 65 535. (Maps via USHORT)

LENGTH4 Length value, in the range 0 through 4 294 967 295.
(Maps via ULONG)

LHANDLE The handle of a resource.

(Scalar type: Maps to **dword**)

LINEBUNDLE Line-attributes bundle structure.
LINEBUNDLE struc

lwnd_lColor	dd	?	;Line foreground color
lwnd_lReserved	dd	?	;Reserved
lwnd_usMixMode	dw	?	;Line foreground-mix mode
lwnd_usReserved	dw	?	;Reserved
lwnd_fxWidth	dd	?	;Line width
lwnd_lGeomWidth	dd	?	;Geometric line width
lwnd_usType	dw	?	;Line type
lwnd_usEnd	dw	?	;Line end
lwnd_usJoin	dw	?	;Line join

LINEBUNDLE ends

LONG Signed integer in the range —2 147 483 648 through 2 147 483 647.

Note: Where this data type represents a graphic coordinate in world or model space, its value is restricted to —134 217 728 through 134 217 727. A graphic coordinate in device or screen coordinates is restricted to —32 768 through 32 767. Its value may be further restricted by any transforms currently in force, including the positioning of the origin of the window on the screen. In particular, coordinates in world or model space must not generate coordinate values after transformation (that is in device or screen space) outside the range —32 768 through 32 767.

(Scalar type: Maps to **dword**)

MARGSTRUCT Multi-line entry field margin information.
MARGSTRUCT struc

m_fsFlags	dw	?
m_usMouMsg	dw	?
m_iptNear	dd	?

MARGSTRUCT ends

MARKERBUNDLE Marker-attributes bundle structure.
MARKERBUNDLE struc

mbnd_lColor	dd	?	;Marker foreground color
mbnd_lBackColor	dd	?	;Marker background color
mbnd_usMixMode	dw	?	;Marker foreground-mix mode
mbnd_usBackMixMode	dw	?	;Marker background-mix mode
mbnd_usSet	dw	?	;Marker set
mbnd_usSymbol	dw	?	;Marker symbol
mbnd_sizfxCell	dw	(size SIZEF)/2 dup (?)	;Marker cell

MARKERBUNDLE ends

MATRIX Matrix-elements structure.
(Maps via MATRIXLF)

MATRIXLF

Matrix-elements structure.

MATRIXLF struc

```

matlf_fxM11 dd ? ;First element of first row
matlf_fxM12 dd ? ;Second element of first row
matlf_lM13 dd ? ;Third element of first row
matlf_fxM21 dd ? ;First element of second row
matlf_fxM22 dd ? ;Second element of second row
matlf_lM23 dd ? ;Third element of second row
matlf_lM31 dd ? ;First element of third row
matlf_lM32 dd ? ;Second element of third row
matlf_lM33 dd ? ;Third element of third row

```

MATRIXLF ends

MENUITEM

Menu item.

MENUITEM struc

```

mi_iPosition dw ? ;Position
mi_afStyle dw ? ;Style
mi_afAttribute dw ? ;Attribute
mi_id dw ? ;Identity
mi_hwndSubMenu dd ? ;Submenu
mi_hItem dd ? ;Item

```

MENUITEM ends

MLECTLDATA

Multi-line entry-field control data structure.

MLECTLDATA struc

```

efd_cbCtlData dw ? ;Length of control data in bytes
efd_afIEFormat dw ? ;Import/export format
efd_cchText dd ? ;Text limit
efd_iptAnchor dd ? ;Selection anchor point
efd_iptCursor dd ? ;Selection cursor point
efd_cxFormat dd ? ;Formatting rectangle width in
pixels
efd_cyFormat dd ? ;Formatting rectangle height in
pixels
efd_afFormatFlags dd ? ;Format flags

```

MLECTLDATA ends

MLE_SEARCHDATA

Search structure for multi-line entry field.

MLE_SEARCHDATA struc

```

se_sStructSize dw ? ;Size of MLE_SEARCHDATA
structure
se_sFindStringLength dw ? ;Length of findstring string
se_pszFindString dd ? ;String to search for
se_iptStart dd ? ;Point at which to start
search, or point where
string was found
se_iptStop dd ? ;Point at which to stop search
se_sFoundStringLength dw ? ;Length of string found at
start
se_sChangeStringLength dw ? ;Length of replacement string
se_pszChangeString dd ? ;Replacement string

```

MLE_SEARCHDATA ends

MPARAM	<p>4-byte message-dependent parameter structure.</p> <p>Certain elements of information, placed into the parameters of a message, have data types that do not consume the whole of the 4 bytes of this data type. The rules governing these cases are:</p> <p>BOOL The value is contained in the low word and the high word is zero.</p> <p>SHORT The value is contained in the low word and its sign is extended into the high word.</p> <p>USHORT The value is contained in the low word and the high word is zero.</p> <p>NULL The entire 4 bytes are zero.</p> <p>(Scalar type: Maps to dword)</p>
MQINFO	<p>Message-queue information structure.</p> <pre> MQINFO struc mqi_cb dw ? ;Length of structure mqi_pid dw ? ;Process identity mqi_tid dw ? ;Thread identity mqi_cmsgs dw ? ;Message count mqi_pReserved dd ? ;Reserved MQINFO ends </pre>
MRESULT	<p>4-byte message-dependent reply parameter structure.</p> <p>Certain elements of information, placed into the parameters of a message, have data types that do not consume the whole of the 4 bytes of this data type. The rules governing these cases are:</p> <p>BOOL The value is contained in the low word and the high word is zero.</p> <p>SHORT The value is contained in the low word and its sign is extended into the high word.</p> <p>USHORT The value is contained in the low word and the high word is zero.</p> <p>NULL The entire 4 bytes are zero.</p> <p>(Scalar type: Maps to dword)</p>
MT	<p>Menu template. (Maps via PVOID)</p>
MTI	<p>Menu template item.</p> <pre> MTI struc mti_afStyle dw ? ;Style mti_afAttrs dw ? ;Attributes mti_idItem dw ? ;Item identity mti_c db 2 dup (?) ;Item data MTI ends </pre>
NPBYTE	<p>Near pointer to a byte string. (Scalar type: Maps to word)</p>
OFFSET2B	<p>Offset value in bytes, in the range 0 through 65 535. (Maps via USHORT)</p>
OVERFLOW	<p>Overflow error structure for multi-line entry field.</p> <pre> OVERFLOW struc ofl_ulErrInd dd ? ;One or more EFR_* flags ofl_lBytesOver dd ? ;Number of bytes over the limit ofl_pixHorzOver dd ? ;Number of pixels over the horizontal limit ofl_pixVertOver dd ? ;Number of pixels over the vertical limit OVERFLOW ends </pre>

OWNERITEM	<p>Owner item.</p> <pre> OWNERITEM struc oi_hwnd dd ? ;Window handle oi_hps dd ? ;Presentation-space handle oi_fsState dw ? ;State oi_fsAttribute dw ? ;Attribute oi_fsStateOld dw ? ;Old state oi_fsAttributeOld dw ? ;Old attribute oi_rclItem dw (size RECTL)/2 dup (?) ;Item rectangle oi_idItem dw ? ;Item identity oi_hItem dd ? ;Item OWNERITEM ends </pre>
PARAM	<p>Presentation parameter attribute definition.</p> <pre> PARAM struc param_id dd ? ;Attribute type identity param_cb dd ? ;Byte count of the attrvalue parameter param_ab db ? ;Attribute value PARAM ends </pre>
PBYTE	<p>Pointer to a data area. (Scalar type: Maps to dword)</p>
PCH	<p>Pointer to a character string. (Scalar type: Maps to dword)</p>
PDEVOPENDATA	<p>Open device-data array.</p> <p>This data type points to data whose format is described by the <i>DEVOPENSTRUC</i> data type.</p> <p>(Scalar type: Maps to dword - address of a DEVOPENDATA)</p>
PFN	<p>Pointer to procedure. (Scalar type: Maps to dword)</p>
PFNWP	<p>Pointer to a window procedure. (Scalar type: Maps to dword)</p>
PIBSTRUCT	<p>Program-information-block structure.</p> <pre> PIBSTRUCT struc pib_progt dw (size PROGTYPE)/2 dup (?) ;Program type and visibility pib_szTitle db (MAXNAME+1) dup (?) ;Program title (null terminated) pib_szIconFileName db (MAXPATHL+1) dup (?) ;Program icon filename (null terminated) pib_szExecutable db (MAXPATHL+1) dup (?) ;Executable file name (null terminated) pib_szStartupDir db (MAXPATHL+1) dup (?) ;Start-up directory (null terminated) pib_xywinInitial dw (size XYWINSIZE)/2 dup (?) ;Initial window position and size pib_res1 dw ? ;Reserved; must be zero pib_res2 dw (size LHANDLE)/2 dup (?) ;Reserved; must be zero pib_cchEnvironmentVars dw ? ;Environment string length pib_pchEnvironmentVars dd ? ;Environment string pib_cchProgramParameter dw ? ;Parameter string length pib_pchProgramParameter dd ? ;Parameter string PIBSTRUCT ends </pre>

PID	Process identity. (Scalar type: Maps to word)
PIX	PeI count for multi-line entry field. (Maps via LONG)
POINT	Point structure. (Maps via POINTL)
POINTERINFO	Pointer-information structure. POINTERINFO struc ptri_fPointer dw ? ;Bit-map size indicator ptri_xHotspot dw ? ;x coordinate of action point ptri_yHotspot dw ? ;y coordinate of action point ptri_hbmPointer dd ? ;Bit-map handle of pointer ptri_hbmColor dd ? ;Bit-map handle of color bit map POINTERINFO ends
POINTL	Point structure (long integer). POINTL struc ptl_x dd ? ;x coordinate ptl_y dd ? ;y coordinate POINTL ends
POINTS	Point structure (short integer). POINTS struc pts_x dw ? ;x coordinate pts_y dw ? ;y coordinate POINTS ends
PQMOPENDATA	Open queue-manager data array. This data type points to data whose format is described by the <i>DEVOPENSTRUC</i> data type. (Scalar type: Maps to dword — address of a QMOPENDATA)
PRESDATA	Class-specific control presentation data, conforming to a <i>PRESPARAM</i> data type. (Maps via PVOID)
PRESPARAMS	Presentation parameter data. PRESPARAMS struc pres_cb dd ? ;Byte count of the param parameter pres_aparam dw (size PARAM)/2 dup (?) ;Array of attribute parameters PRESPARAMS ends
PRFPROFILE	Profile structure. PRFPROFILE struc prfpro_cchUserName dd ? ;Length of user profile name prfpro_pszUserName dd ? ;User profile name prfpro_cchSysName dd ? ;Length of system profile name prfpro_pszSysName dd ? ;System profile name PRFPROFILE ends
PROC	Procedure identifier. (Maps via PFN)
PROGCATEGORY	Program category. (Scalar type: Maps to byte)

PROGDETAILS

Program details structure.

```

PROGDETAILS struc
    progde_Length      dd ?      ;Length of structure
    progde_progt       dw (size PROGTYPE)/2 dup (?) ;Program
                                           type
    progde_pad1        dw 3 dup (?) ;Reserved
    progde_pszTitle     dd ?      ;Title
    progde_pszExecutable dd ?      ;Executable file name
    progde_pszParameters dd ?      ;Parameter string
    progde_pszStartupDir dd ?      ;Start-up directory
    progde_pszIcon      dd ?      ;Icon-file name
    progde_pszEnvironment dd ?      ;Environment string. (List of
                                           null-terminated strings,
                                           ending with an extra null)
    progde_swpInitial   dw (size SWP)/2 dup (?) ;Initial window
                                           position and size
    progde_pad2        dw 5 dup (?) ;Reserved
PROGDETAILS ends

```

PROGRAMENTRY

Program-entry structure.

```

PROGRAMENTRY struc
    proge_hprog dd ?      ;Program handle
    proge_progt dw (size PROGTYPE)/2 dup (?) ;Program type
    proge_szTitle db (MAXNAME+1) dup (?) ;Program title
                                           (null terminated)
PROGRAMENTRY ends

```

PROGTITLE

Program-title structure.

```

PROGTITLE struc
    progti_hprog dd ?      ;Program handle
    progti_progt dw (size PROGTYPE)/2 dup (?) ;Program type
    progti_pad1 dw 3 dup (?) ;Reserved
    progti_pszTitle dd ?      ;Program title
PROGTITLE ends

```

PROGTYPE

Program-type structure.

```

PROGTYPE struc
    progt_progc db ?      ;Program category
    progt_fbVisible db ?      ;Visibility attribute
PROGTYPE ends

```

PROPERTY2Property value. Up to 65 536 different properties are available.
(Maps via USHORT)**PROPERTY4**Property value. Up to 4 294 967 296 different properties are available.
(Maps via LONG)**PSZ**Pointer to a null-terminated string.
(Scalar type: Maps to **dword** — address of an ASCIIZ)**PVOID**Pointer to a data type of undefined format.
(Scalar type: Maps to **dword**)**QMOPENDATA**Open queue-manager data area. The format of this area is the same as a *DEVOPENSTRUC* structure.
(Maps via PQMOPENDATA)

QMSG	<p>Message structure.</p> <pre> QMSG struc qmsg_hwnd dd ? ;Window handle qmsg_msg dw ? ;Message identity qmsg_mp1 dd ? ;Parameter 1 qmsg_mp2 dd ? ;Parameter 2 qmsg_time dd ? ;Message time qmsg_ptl dw (size POINTL)/2 dup (?) ;Pointer position when message was generated QMSG ends </pre>
RECT	<p>Rectangle structure.</p> <p>Unless otherwise stated, points on the right-hand or top boundaries of the rectangle are not considered to be included in the rectangle; points on the left-hand or bottom boundaries (that are not also on the right-hand or top boundaries) are included in the rectangle. (Maps via RECTL)</p>
RECTL	<p>Rectangle structure.</p> <pre> RECTL struc rcl_xLeft dd ? ;x coordinate of left-hand edge of rectangle rcl_yBottom dd ? ;y coordinate of bottom edge of rectangle rcl_xRight dd ? ;x coordinate of right-hand edge of rectangle rcl_yTop dd ? ;y coordinate of top edge of rectangle RECTL ends </pre>
RESID	<p>Resource identity. (Maps via HMODULE)</p>
RGB	<p>RGB color value.</p> <pre> RGB struc rgb_bBlue db ? ;Blue component of the color definition rgb_bGreen db ? ;Green component of the color definition rgb_bRed db ? ;Red component of the color definition RGB ends </pre>
RGNRECT	<p>Region-rectangle structure.</p> <pre> RGNRECT struc rgnrc_ircStart dw ? ;Rectangle number from which to start enumerating rgnrc_crc dw ? ;Number of rectangles that can be returned rgnrc_crcReturned dw ? ;Number of rectangles returned rgnrc_usDirection dw ? ;Direction in which the returned rectangles are to be ordered RGNRECT ends </pre>
ROF	<p>Number representation, equivalent to <i>FIXED</i>. (Maps via FIXED)</p>
ROL	<p>Number representation, equivalent to <i>LONG</i>. (Maps via LONG)</p>

SBCDATA	<p>Scroll-bar control data structure.</p> <pre> SBCDATA struc sbcd_cb dw ? ;Length of control data in bytes sbcd_sHilite dw ? ;Highlighting code sbcd_posFirst dw ? ;First bound of the scroll-bar range sbcd_posLast dw ? ;Last bound of the scroll-bar range sbcd_posThumb dw ? ;Slider position sbcd_cVisible dw ? ;Number of data items visible sbcd_cTotal dw ? ;Number of data items available SBCDATA ends </pre>
SEGOFF	<p>2-byte segment offset in bytes. (Maps via NPBYTE)</p>
SHORT	<p>Signed integer in the range —32 768 through 32 767. (Scalar type: Maps to word)</p>
SIZEF	<p>Size structure.</p> <pre> SIZEF struc sizfx_cx dd ? ;Width sizfx_cy dd ? ;Height SIZEF ends </pre>
SIZEL	<p>Size structure.</p> <pre> SIZEL struc sizl_cx dd ? ;Width sizl_cy dd ? ;Height SIZEL ends </pre>
SIZEROF	<p>Size structure. (Maps via SIZEF)</p>
SIZEROL	<p>Size structure. (Maps via SIZEL)</p>
SMHSTRUCT	<p>Send message hook structure.</p> <pre> SMHSTRUCT struc smhs_mp2 dd ? ;Parameter 2 smhs_mp1 dd ? ;Parameter 1 smhs_msg dw ? ;Message identity smhs_hwnd dd ? ;Window handle SMHSTRUCT ends </pre>
STORAGE	<p>Storage that may contain different data types. (Maps via PVOID)</p>
STR	<p>String with an explicit length count. (Maps via PCH)</p>
STRCOND	<p>String with a length count that is determined by special values of its associated length parameter.</p> <p>For example, positive values of the associated length parameter provide the length count explicitly, whereas the value —1 implies that the string is null-terminated and hence its length is implicit.</p> <p>Where this data type occurs, the associated length parameter specifies the meanings of its values in determining the length count of the string. (Maps via PCH)</p>

TIME

Time interval in milliseconds.
(Maps via LONG)

TRACKINFO

Tracking-information structure.
TRACKINFO struc
ti_cxBorder dw ? ;Border width
ti_cyBorder dw ? ;Border height
ti_cxGrid dw ? ;Grid width
ti_cyGrid dw ? ;Grid height
ti_cxKeyboard dw ? ;Character cell width movement for
arrow key
ti_cyKeyboard dw ? ;Character cell height movement for
arrow key
ti_rclTrack dw (size RECTL)/2 dup (?) ;Starting tracking
rectangle
ti_rclBoundary dw (size RECTL)/2 dup (?) ;Boundary
rectangle
ti_ptlMinTrackSize dw (size POINTL)/2 dup (?) ;Minimum
tracking size
ti_ptlMaxTrackSize dw (size POINTL)/2 dup (?) ;Maximum
tracking size
ti_fs dw ? ;Tracking options
TRACKINFO ends

UCHAR

Unsigned integer in the range 0 through 255.
(Scalar type: Maps to byte)

ULONG

Unsigned integer in the range 0 through 4 294 967 295.
(Scalar type: Maps to dword)

USERBUTTON

User-button data structure.
USERBUTTON struc
ubtn_hwnd dd ? ;Window handle
ubtn_hps dd ? ;Presentation-space handle
ubtn_fsState dw ? ;New state of user button
ubtn_fsStateOld dw ? ;Old state of user button
USERBUTTON ends

USHORT

Unsigned integer in the range 0 through 65 535.
(Scalar type: Maps to word)

VOID

A data area of undefined format.
(Scalar type: Maps to other)

WIDTH4

Width in the range -2 147 483 648 through 2 147 483 647.
(Maps via LONG)

WNDPARAMS

Window parameters.
WNDPARAMS struc
wprm_fsStatus dw ? ;Window parameter selection
wprm_cchText dw ? ;Length of window text
wprm_pszText dd ? ;Window text
wprm_cbPresParams dw ? ;Length of presentation parameters
wprm_pPresParams dd ? ;Presentation parameters
wprm_cbCtlData dw ? ;Length of window class specific data
wprm_pCtlData dd ? ;Window class specific data
WNDPARAMS ends

WNDPROC

Window-procedure identifier.
(Maps via PFNWP)

WPOINT

Window point structure (integer).

```
WPOINT struc
    wpt_x      dw    ?    ;x coordinate
    wpt_dummy1 dw    ?    ;Reserved
    wpt_y      dw    ?    ;y coordinate
    wpt_dummy2 dw    ?    ;Reserved
WPOINT ends
```

WRECT

Window rectangle structure (integer).

```
WRECT struc
    wrc_xLeft  dw    ?    ;x coordinate of left-hand edge of
                           rectangle
    wrc_dummy1 dw    ?    ;Reserved
    wrc_yBottom dw    ?    ;y coordinate of bottom edge of rectangle
    wrc_dummy2 dw    ?    ;Reserved
    wrc_xRight dw    ?    ;x coordinate of right-hand edge of
                           rectangle
    wrc_dummy3 dw    ?    ;Reserved
    wrc_yTop   dw    ?    ;y coordinate of top edge of rectangle
    wrc_dummy4 dw    ?    ;Reserved
WRECT ends
```

XYWINSIZE

Window position and size structure.

```
XYWINSIZE struc
    xywin_x    dw    ?    ;x coordinate of window origin
    xywin_y    dw    ?    ;y coordinate of window origin
    xywin_cx   dw    ?    ;Window width
    xywin_cy   dw    ?    ;Window height
    xywin_fsWindow dw    ?    ;Window options
XYWINSIZE ends
```

Chapter 3. Device Function Calls

DevCloseDC

```
EXTRN DevCloseDC:FAR
INCL_DEV equ 1 ;Or use INCL_PM. Also in COMMON section

PUSH DWORD hdc ;Device-context handle
CALL DevCloseDC

Returns DWORD hmf ;Error indicator/metafile handle (for a metafile device context)
```

DevEscape

```
EXTRN DevEscape:FAR
INCL_DEV equ 1 ;Or use INCL_PM

PUSH DWORD hdc ;Device-context handle
PUSH DWORD lCode ;Escape code
PUSH DWORD lInCount ;Input data count
PUSH@ OTHER bInData ;The input data required for this escape
PUSH@ DWORD lOutCount ;Output data count
PUSH@ OTHER bOutData ;Output data
CALL DevEscape

Returns DWORD lResult ;Implemented/error indicator
```

DevOpenDC

```
EXTRN DevOpenDC:FAR
INCL_DEV equ 1 ;Or use INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD lType ;Type of device context
PUSH@ ASCIIZ szToken ;Device-information token
PUSH DWORD lCount ;Number of items
PUSH@ OTHER dopData ;Open-device-context data area
PUSH DWORD hdcComp ;Compatible-device-context handle
CALL DevOpenDC

Returns DWORD hdc ;Device-context handle
```

DevPostDeviceModes

```
EXTRN DevPostDeviceModes:FAR
INCL_DEV equ 1 ;Or use INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER drivDriverData ;Driver data
PUSH@ ASCIIZ szDriverName ;This is a string containing the name of the presentation
                        device driver
PUSH@ ASCIIZ szDeviceName ;Device-type name
PUSH@ ASCIIZ szName ;Device name
PUSH DWORD fOptions ;Dialog options
CALL DevPostDeviceModes

Returns DWORD lDriverCount ;Size/error indicator
```

DevQueryCaps

```
EXTRN DevQueryCaps:FAR
INCL_DEV equ 1 ;Or use INCL_PM. Also in COMMON section

PUSH  DWORD   hdc           ;Device-context handle
PUSH  DWORD   lStart        ;First item of information
PUSH  DWORD   lCount        ;Count of items of information
PUSH@ OTHER   alArray       ;Device capabilities
CALL  DevQueryCaps

Returns WORD   fSuccess ;Success indicator
```

DevQueryDeviceNames

```
EXTRN DevQueryDeviceNames:FAR
INCL_DEV equ 1 ;Or use INCL_PM

PUSH  DWORD   hab           ;Anchor-block handle
PUSH@ ASCIIZ   szDriverName ;Fully qualified name of the file containing the presentation
                                driver
PUSH@ DWORD    ldn           ;Maximum number of device names and descriptions that can be
                                returned
PUSH@ OTHER    aDeviceName   ;Device-name array
PUSH@ OTHER    aDeviceDesc   ;Device description array
PUSH@ DWORD    ldt           ;Maximum number of data types that can be returned
PUSH@ OTHER    aDataType     ;Data type array
CALL  DevQueryDeviceNames

Returns WORD   fSuccess ;Success indicator
```

DevQueryHardcopyCaps

```
EXTRN DevQueryHardcopyCaps:FAR
INCL_DEV equ 1 ;Or use INCL_PM

PUSH  DWORD   hdc           ;Device-context handle
PUSH  DWORD   lStartForm     ;Start-forms code
PUSH  DWORD   lForms         ;Number of forms to query
PUSH@ OTHER    hciHcInfo     ;Hardcopy capabilities information
CALL  DevQueryHardcopyCaps

Returns DWORD   lFormsReturned ;Details of forms
```

Chapter 4. Graphics Presentation Interface Function Calls

GpiAssociate

```
EXTRN GpiAssociate:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD hdc ;Device-context handle
CALL GpiAssociate

Returns WORD fSuccess ;Success indicator
```

GpiBeginArea

```
EXTRN GpiBeginArea:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD fOptions ;Area options
CALL GpiBeginArea

Returns WORD fSuccess ;Success indicator
```

GpiBeginElement

```
EXTRN GpiBeginElement:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lType ;Type to be associated with element
PUSH@ ASCIIZ szDesc ;Description
CALL GpiBeginElement

Returns WORD fSuccess ;Success indicator
```

GpiBeginPath

```
EXTRN GpiBeginPath:FAR
INCL_GPIPATHS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lPath ;Path identifier
CALL GpiBeginPath

Returns WORD fSuccess ;Success indicator
```

GpiBitBlt

```
EXTRN GpiBitBlt:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hpsTarget ;Target presentation-space handle
PUSH DWORD hpsSource ;Source presentation-space handle
PUSH DWORD lCount ;Point count
PUSH@ OTHER aptlPoints ;Point array
PUSH DWORD lRop ;Mixing function required
PUSH DWORD fOptions ;Options
CALL GpiBitBlt

Returns DWORD lHits ;Correlation/error indicator
```

GpiBox

```
EXTRN GpiBox:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lControl ;Outline and fill control
PUSH@ OTHER  ptlPoint ;Corner point
PUSH  DWORD  lHRound  ;Corner-rounding control
PUSH  DWORD  lVRound  ;Corner-rounding control
CALL  GpiBox

Returns DWORD  lHits    ;Correlation/error indicator
```

GpiCallSegmentMatrix

```
EXTRN GpiCallSegmentMatrix:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lSegment ;Identifier of segment to be called
PUSH  DWORD  lCount   ;Number of elements
PUSH@ OTHER  matlfArray ;Instance transform matrix
PUSH  DWORD  lOptions ;Transformation options
CALL  GpiCallSegmentMatrix

Returns DWORD  lHits    ;Correlation/error indicator
```

GpiCharString

```
EXTRN GpiCharString:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Number of characters
PUSH@ OTHER  chString ;Characters to be drawn
CALL  GpiCharString

Returns DWORD  lHits    ;Correlation/error indicator
```

GpiCharStringAt

```
EXTRN GpiCharStringAt:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  ptlPoint ;Starting position
PUSH  DWORD  lCount   ;Number of characters
PUSH@ OTHER  chString ;Characters to be drawn
CALL  GpiCharStringAt

Returns DWORD  lHits    ;Correlation/error indicator
```

GpiCharStringPos

```
EXTRN GpiCharStringPos:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  rc1Rect  ;Rectangle structure
PUSH  DWORD  flOptions ;Formatting options
PUSH  DWORD  lCount   ;Number of characters in the string
PUSH@ OTHER  chString ;Character string
PUSH@ OTHER  alAdx    ;Increment values
CALL  GpiCharStringPos

Returns DWORD  lHits    ;Correlation/error indicator
```

GpiCharStringPosAt

```
EXTRN GpiCharStringPosAt:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptlStart ;Starting position
PUSH@ OTHER rclRect ;Rectangle structure
PUSH DWORD flOptions ;Formatting options
PUSH DWORD lCount ;Number of characters in the string
PUSH@ OTHER chString ;Character string
PUSH@ OTHER alAdx ;Increment values
CALL GpiCharStringPosAt

Returns DWORD lHits ;Correlation/error indicator
```

GpiCloseFigure

```
EXTRN GpiCloseFigure:FAR
INCL_GPIPATHS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiCloseFigure

Returns WORD fSuccess ;Success indicator
```

GpiCloseSegment

```
EXTRN GpiCloseSegment:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiCloseSegment

Returns WORD fSuccess ;Success indicator
```

GpiCombineRegion

```
EXTRN GpiCombineRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD hrgnDest ;Destination-region handle
PUSH DWORD hrgnSrc1 ;First-source-region handle
PUSH DWORD hrgnSrc2 ;Second-source-region handle
PUSH DWORD lMode ;Method of combination
CALL GpiCombineRegion

Returns DWORD lComplexity ;Complexity of resulting region/error indicator
```

GpiComment

```
EXTRN GpiComment:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lLength ;Data length
PUSH@ OTHER bData ;Comment data
CALL GpiComment

Returns WORD fSuccess ;Success indicator
```

GpiConvert

```
EXTRN GpiConvert:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lSrc     ;Source-coordinate space
PUSH  DWORD  lTarg    ;Target-coordinate space
PUSH  DWORD  lCount   ;Point count
PUSH@ OTHER  aptlPoints ;Array of (x,y) coordinate pair structures
CALL  GpiConvert

Returns WORD  fSuccess ;Success indicator
```

GpiCopyMetaFile

```
EXTRN GpiCopyMetaFile:FAR
INCL_GPIMETAFILES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hmf      ;Source metafile handle
CALL  GpiCopyMetaFile

Returns DWORD  hmfNew ;New metafile handle/error indicator
```

GpiCorrelateChain

```
EXTRN GpiCorrelateChain:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lType     ;Segment type
PUSH@ OTHER  ptlPick   ;Pick position
PUSH  DWORD  lMaxHits  ;Maximum hits
PUSH  DWORD  lMaxDepth ;Number of pairs
PUSH@ OTHER  alSegTag  ;Segment identifiers and tags
CALL  GpiCorrelateChain

Returns DWORD  lNumHits ;Number of hits or error
```

GpiCorrelateFrom

```
EXTRN GpiCorrelateFrom:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lFirstSegment ;Specifies the first segment to be correlated
PUSH  DWORD  lLastSegment ;Specifies the last segment to be correlated
PUSH  DWORD  lType     ;The type of segments on which correlation is to be performed
PUSH@ OTHER  ptlPick   ;Pick position
PUSH  DWORD  lMaxHits  ;Maximum hits
PUSH  DWORD  lMaxDepth ;Number of pairs
PUSH@ OTHER  alSegTag  ;An array of segment and tag identifiers in alternate elements
CALL  GpiCorrelateFrom

Returns DWORD  lNumHits ;Number of hits, or error
```

GpiCorrelateSegment

```
EXTRN GpiCorrelateSegment:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lSegment ;Identifier of the segment to be correlated; it must be greater than
                        0
PUSH  DWORD  lType    ;Type of segments on which correlation is to be performed
PUSH@  OTHER  ptlPick  ;Position of the center of the pick aperture in presentation page
                        space
PUSH  DWORD  lMaxHits ;Maximum hits
PUSH  DWORD  lMaxDepth ;Number of segment and tag pairs to be returned for each hit
PUSH@  OTHER  alSegTag ;Array
CALL  GpiCorrelateSegment

Returns DWORD  lNumHits ;Number of hits, or error
```

GpiCreateBitmap

```
EXTRN GpiCreateBitmap:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@  OTHER  bmpNew   ;Bit-map information header
PUSH  DWORD  flOptions ;Options
PUSH@  OTHER  bInitData ;Buffer address
PUSH@  OTHER  bmiInfoTable ;Bit-map information table
CALL  GpiCreateBitmap

Returns DWORD  hbm      ;Bit-map handle/error indicator
```

GpiCreateLogColorTable

```
EXTRN GpiCreateLogColorTable:FAR
INCL_GPILOGCOLORTABLE equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  flOptions ;Options
PUSH  DWORD  lFormat   ;Format of entries in the table
PUSH  DWORD  lStart    ;Starting index
PUSH  DWORD  lCount    ;Count of elements in Table
PUSH@  OTHER  alTable  ;Start of the application data area
CALL  GpiCreateLogColorTable

Returns WORD  fSuccess ;Success indicator
```

GpiCreateLogFont

```
EXTRN GpiCreateLogFont:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@  OTHER  Name     ;Logical font name
PUSH  DWORD  lLcid     ;Local identifier
PUSH@  OTHER  fatAttrs ;Attributes of font
CALL  GpiCreateLogFont

Returns DWORD  lMatch  ;Match indicator
```


GpiCreatePS

```
EXTRN GpiCreatePS:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hdc ;Device-context handle
PUSH@ OTHER sizlSize ;Presentation-page size
PUSH DWORD flOptions ;Options
CALL GpiCreatePS

Returns DWORD hps ;Presentation-space handle
```

GpiCreateRegion

```
EXTRN GpiCreateRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lCount ;The number of rectangles
PUSH@ OTHER arclRectangles ;An array of rectangles
CALL GpiCreateRegion

Returns DWORD hrgn ;Region handle
```

GpiDeleteBitmap

```
EXTRN GpiDeleteBitmap:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hbm ;Handle of bit map to be deleted
CALL GpiDeleteBitmap

Returns WORD fSuccess ;Success indicator
```

GpiDeleteElement

```
EXTRN GpiDeleteElement:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiDeleteElement

Returns WORD fSuccess ;Success indicator
```

GpiDeleteElementRange

```
EXTRN GpiDeleteElementRange:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lFirstElement ;Number of the first element to be deleted
PUSH DWORD lLastElement ;Number of the last element to be deleted
CALL GpiDeleteElementRange

Returns WORD fSuccess ;Success indicator
```

GpiDeleteElementsBetweenLabels

```
EXTRN GpiDeleteElementsBetweenLabels:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lFirstLabel ;Label marking the start of the elements to be deleted
PUSH DWORD lLastLabel ;Label marking the end of the elements to be deleted
CALL GpiDeleteElementsBetweenLabels

Returns WORD fSuccess ;Success indicator
```

GpiDeleteMetaFile

```
EXTRN GpiDeleteMetaFile:FAR
INCL_GPIMETAFILES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hmf ;Metafile handle
CALL GpiDeleteMetaFile

Returns WORD fSuccess ;Success indicator
```

GpiDeleteSegment

```
EXTRN GpiDeleteSegment:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lSegid ;Segment identifier
CALL GpiDeleteSegment

Returns WORD fSuccess ;Success indicator
```

GpiDeleteSegments

```
EXTRN GpiDeleteSegments:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lFirstSegment ;First identifier in the range; it must be greater than 0
PUSH DWORD lLastSegment ;Last identifier in the range; it must be greater than 0
CALL GpiDeleteSegments

Returns WORD fSuccess ;Success indicator
```

GpiDeleteSetId

```
EXTRN GpiDeleteSetId:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lLcid ;Local identifier
CALL GpiDeleteSetId

Returns WORD fSuccess ;Success indicator
```

GpiDestroyPS

```
EXTRN GpiDestroyPS:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
CALL GpiDestroyPS

Returns WORD fSuccess ;Success indicator
```

GpiDestroyRegion

```
EXTRN GpiDestroyRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD hrgn ;Handle of region to be destroyed
CALL GpiDestroyRegion

Returns WORD fSuccess ;Success indicator
```

GpiDrawChain

```
EXTRN GpiDrawChain:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiDrawChain

Returns WORD fSuccess ;Success indicator
```

GpiDrawDynamics

```
EXTRN GpiDrawDynamics:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiDrawDynamics

Returns WORD fSuccess ;Success indicator
```

GpiDrawFrom

```
EXTRN GpiDrawFrom:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lFirstSegment ;First segment to be drawn; it must be greater than zero
PUSH DWORD lLastSegment ;Last segment to be drawn; it must be greater than zero
CALL GpiDrawFrom

Returns WORD fSuccess ;Success indicator
```

GpiDrawSegment

```
EXTRN GpiDrawSegment:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lSegment ;Segment to be drawn; it must be greater than zero
CALL GpiDrawSegment

Returns WORD fSuccess ;Success indicator
```

GpiElement

```
EXTRN GpiElement:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lType ;Type to be associated with the element
PUSH@ ASCIIZ szDesc ;Element description
PUSH DWORD lLength ;Length of content data for the element
PUSH@ OTHER bData ;Buffer pointer
CALL GpiElement

Returns DWORD lHits ;Correlation/error indicator
```

GpiEndArea

```
EXTRN GpiEndArea:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
CALL GpiEndArea

Returns DWORD lHits ;Correlation/error indicator
```

GpiEndElement

```
EXTRN GpiEndElement:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiEndElement

Returns WORD fSuccess ;Success indicator
```

GpiEndPath

```
EXTRN GpiEndPath:FAR
INCL_GPIPATHS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiEndPath

Returns WORD fSuccess ;Success indicator
```

GpiEqualRegion

```
EXTRN GpiEqualRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD hrgnSrc1 ;Handle of first region
PUSH DWORD hrgnSrc2 ;Handle of second region
CALL GpiEqualRegion

Returns DWORD lEquality ;Equality/error indicator
```

GpiErase

```
EXTRN GpiErase:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
CALL GpiErase

Returns WORD fSuccess ;Success indicator
```

GpiErrorSegmentData

```
EXTRN GpiErrorSegmentData:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ DWORD lSegment ;Segment in which the error occurred
PUSH@ DWORD lContext ;Context of the error
CALL GpiErrorSegmentData

Returns DWORD lOff ;Position
```

GpiExcludeClipRectangle

```
EXTRN GpiExcludeClipRectangle:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rclRectangle ;Rectangle to be excluded
CALL GpiExcludeClipRectangle

Returns DWORD lComplexity ;Complexity of clipping/error indicator
```

GpiFillPath

```
EXTRN GpiFillPath:FAR
INCL_GPIPATHS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lPath ;Identifier of path whose interior is to be drawn; it must be 1
PUSH DWORD lOptions ;Fill option
CALL GpiFillPath

Returns DWORD lHits ;Error indicator
```

GpiFullArc

```
EXTRN GpiFullArc:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lControl ;Interior/outline control
PUSH DWORD fxMultiplier ;Multiplier
CALL GpiFullArc

Returns DWORD lHits ;Correlation/error indicator
```

GpiGetData

```
EXTRN GpiGetData:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lSegid ;Segment identifier
PUSH@ DWORD lOffset ;Segment offset
PUSH DWORD lFormat ;Coordinate type required
PUSH DWORD lLength ;Length of data buffer
PUSH@ OTHER bData ;Data buffer
CALL GpiGetData

Returns DWORD lCount ;Length of returned data
```

GpiImage

```
EXTRN GpiImage:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lFormat ;Format of image data
PUSH@ OTHER sizlImageSize ;Size of image area (in pels)
PUSH DWORD lLength ;Length in bytes of image data
PUSH@ OTHER bData ;Image data
CALL GpiImage

Returns DWORD lHits ;Correlation/error indicator
```

GpiIntersectClipRectangle

```
EXTRN GpiIntersectClipRectangle:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rcIRectangle ;Rectangle, the coordinates of which are world coordinates
CALL GpiIntersectClipRectangle

Returns DWORD lComplexity ;Complexity of clipping/error indicator
```

GpiLabel

```
EXTRN GpiLabel:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ DWORD lLabel ;Required label
CALL GpiLabel

Returns WORD fSuccess ;Success indicator
```

GpiLine

```
EXTRN GpiLine:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptlEndPoint ;End point of the line
CALL GpiLine

Returns DWORD lHits ;Correlation/error indicator
```

GpiLoadBitmap

```
EXTRN GpiLoadBitmap:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
PUSH WORD Resource ;Resource identity containing the bit map
PUSH WORD idBitmap ;ID of the bit map within the resource file
PUSH DWORD lWidth ;Width of the bit map in pels
PUSH DWORD lHeight ;Height of the bit map in pels
CALL GpiLoadBitmap

Returns DWORD hbm ;Bit-map handle
```

GpiLoadFonts

```
EXTRN GpiLoadFonts:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCII szFilename ;Filename
CALL GpiLoadFonts

Returns WORD fSuccess ;Success indicator
```

GpiLoadMetaFile

```
EXTRN GpiLoadMetaFile:FAR
INCL_GPIMETAFILES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCII szFilename ;Filename
CALL GpiLoadMetaFile

Returns DWORD hmf ;Metafile handle or error
```

GpiMarker

```
EXTRN GpiMarker:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptlPoint ;Position of the marker
CALL GpiMarker

Returns DWORD lHits ;Correlation/error indicator
```

GpiModifyPath

```
EXTRN GpiModifyPath:FAR
INCL_GPIPATHS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lPath ;Path identifier
PUSH DWORD lMode ;Modification required
CALL GpiModifyPath

Returns WORD fSuccess ;Success indicator
```

GpiMove

```
EXTRN GpiMove:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptlPoint ;Position to which to move
CALL GpiMove

Returns WORD fSuccess ;Success indicator
```

GpiOffsetClipRegion

```
EXTRN GpiOffsetClipRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptlPoint ;Displacement
CALL GpiOffsetClipRegion

Returns DWORD lComplexity ;Complexity of clipping/error indicator
```

GpiOffsetElementPointer

```
EXTRN GpiOffsetElementPointer:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lOffset ;Offset to be added to the element pointer
CALL GpiOffsetElementPointer

Returns WORD fSuccess ;Success indicator
```

GpiOffsetRegion

```
EXTRN GpiOffsetRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD Hrgn ;Region handle to be moved
PUSH@ OTHER ptlOffset ;Offset to be added to the region boundary
CALL GpiOffsetRegion

Returns WORD fSuccess ;Success indicator
```

GpiOpenSegment

```
EXTRN GpiOpenSegment:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lSegment ;Segment identifier
CALL GpiOpenSegment

Returns WORD fSuccess ;Success indicator
```

GpiOutlinePath

```
EXTRN GpiOutlinePath:FAR
INCL_GPIPATHS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lPath ;Identifier of path to be outlined; it must be 1
PUSH DWORD lOptions ;Options
CALL GpiOutlinePath

Returns DWORD lHits ;Correlation/error indicator
```

GpiPaintRegion

```
EXTRN GpiPaintRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD hrgn ;Region handle
CALL GpiPaintRegion

Returns DWORD lHits ;Correlation/error indicator
```

GpiPartialArc

```
EXTRN GpiPartialArc:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptlCenter ;Center point
PUSH DWORD fxMultiplier ;Multiplier
PUSH DWORD fxStartAngle ;Start angle in degrees
PUSH DWORD fxSweepAngle ;Sweep angle in degrees
CALL GpiPartialArc

Returns DWORD lHits ;Correlation/error indicator
```

GpiPlayMetaFile

```
EXTRN GpiPlayMetaFile:FAR
INCL_GPIMETAFILES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD hmf ;Metafile handle
PUSH DWORD lCount1 ;Count of elements in Optarray
PUSH@ OTHER alOptarray ;Array of options for playing
PUSH@ DWORD lSegCount ;Reserved
PUSH DWORD lCount2 ;Count of bytes in Desc
PUSH@ ASCIIZ szDesc ;Descriptive record
CALL GpiPlayMetaFile

Returns DWORD lHits ;Correlation/error indicator
```

GpiPointArc

```
EXTRN GpiPointArc:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER aptlPoints ;Intermediate and end points
CALL GpiPointArc

Returns DWORD lHits ;Correlation/error indicator
```


GpiPolyFillet

```
EXTRN GpiPolyFillet:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Number of points
PUSH@ OTHER  aptlPoints ;Array of points
CALL  GpiPolyFillet

Returns DWORD lHits      ;Correlation/error indicator
```

GpiPolyFilletSharp

```
EXTRN GpiPolyFilletSharp:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Count of points
PUSH@ OTHER  aptlPoints ;An array of points
PUSH@ OTHER  afxSharpness ;Array of sharpness values
CALL  GpiPolyFilletSharp

Returns DWORD lHits      ;Correlation/error indicator
```

GpiPolyLine

```
EXTRN GpiPolyLine:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Number of points
PUSH@ OTHER  aptlPoints ;Array of points
CALL  GpiPolyLine

Returns DWORD lHits      ;Correlation/error indicator
```

GpiPolyMarker

```
EXTRN GpiPolyMarker:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Number of points
PUSH@ OTHER  aptlPoints ;Array of points
CALL  GpiPolyMarker

Returns DWORD lHits      ;Correlation/error indicator
```

GpiPolySpline

```
EXTRN GpiPolySpline:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Count of points
PUSH@ OTHER  aptlPoints ;An array of points
CALL  GpiPolySpline

Returns DWORD lHits      ;Correlation/error indicator
```

GpiPop

```
EXTRN GpiPop:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Number of attributes to be restored
CALL  GpiPop

Returns WORD fSuccess ;Success indicator
```

GpiPtInRegion

```
EXTRN GpiPtInRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  hrgn     ;Region handle
PUSH@ OTHER  ptlPoint ;Point to be checked
CALL  GpiPtInRegion

Returns DWORD  lInside ;Inside/error indicator
```

GpiPtVisible

```
EXTRN GpiPtVisible:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  ptlPoint ;Point to be checked
CALL  GpiPtVisible

Returns DWORD  lVisibility ;Visibility indicator
```

GpiPutData

```
EXTRN GpiPutData:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lFormat  ;Coordinate type used
PUSH@ DWORD  lLength  ;Length of graphic data
PUSH@ OTHER  bData    ;Orders to be copied
CALL  GpiPutData

Returns DWORD  lHits   ;Correlation/error indicator
```

GpiQueryArcParams

```
EXTRN GpiQueryArcParams:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  arcpArcParams ;Arc parameters
CALL  GpiQueryArcParams

Returns WORD  fSuccess ;Success indicator
```

GpiQueryAttrMode

```
EXTRN GpiQueryAttrMode:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
CALL  GpiQueryAttrMode

Returns DWORD  lMode   ;Current attribute mode
```

GpiQueryAttrs

```
EXTRN GpiQueryAttrs:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lPrimType ;Primitive type
PUSH  DWORD  flAttrMask ;Attributes mask
PUSH@ OTHER  pbunAttrs ;Attributes
CALL  GpiQueryAttrs

Returns DWORD  lDefMask ;Defaults mask
```

GpiQueryBackColor

```
EXTRN GpiQueryBackColor:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryBackColor

Returns DWORD lColor ;Background color
```

GpiQueryBackMix

```
EXTRN GpiQueryBackMix:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryBackMix

Returns DWORD lMixMode ;Background mix
```

GpiQueryBitmapBits

```
EXTRN GpiQueryBitmapBits:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lScanStart ;Starting line number
PUSH DWORD lScans ;Number of scan lines to be returned
PUSH@ OTHER pBuffer ;Data area
PUSH@ OTHER bmiInfoTable ;Bit-map information table
CALL GpiQueryBitmapBits

Returns DWORD lScansReturned ;Number of scan lines actually returned
```

GpiQueryBitmapDimension

```
EXTRN GpiQueryBitmapDimension:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hbm ;Bit-map handle
PUSH@ OTHER sizlBitmapDimension ;Size of bit map
CALL GpiQueryBitmapDimension

Returns WORD fSuccess ;Success indicator
```

GpiQueryBitmapHandle

```
EXTRN GpiQueryBitmapHandle:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lLcid ;Local identifier
CALL GpiQueryBitmapHandle

Returns DWORD hbm ;Bit-map handle
```

GpiQueryBitmapParameters

```
EXTRN GpiQueryBitmapParameters:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hbm ;Bit-map handle
PUSH@ OTHER bmpData ;Bit-map information header
CALL GpiQueryBitmapParameters

Returns WORD fSuccess ;Success indicator
```

GpiQueryBoundaryData

```
EXTRN GpiQueryBoundaryData:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rcIBoundary ;Boundary data
CALL GpiQueryBoundaryData

Returns WORD fSuccess ;Success indicator
```

GpiQueryCharAngle

```
EXTRN GpiQueryCharAngle:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER gradIAngle ;Baseline angle
CALL GpiQueryCharAngle

Returns WORD fSuccess ;Success indicator
```

GpiQueryCharBox

```
EXTRN GpiQueryCharBox:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER sizfxSize ;Character-box size
CALL GpiQueryCharBox

Returns WORD fSuccess ;Success indicator
```

GpiQueryCharDirection

```
EXTRN GpiQueryCharDirection:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryCharDirection

Returns DWORD lDirection ;Character direction
```

GpiQueryCharMode

```
EXTRN GpiQueryCharMode:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryCharMode

Returns DWORD lMode ;Character mode
```

GpiQueryCharSet

```
EXTRN GpiQueryCharSet:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryCharSet

Returns DWORD lLcid ;Character-set local identifier
```

GpiQueryCharShear

```
EXTRN GpiQueryCharShear:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptIShear ;Character shear
CALL GpiQueryCharShear

Returns WORD fSuccess ;Success indicator
```

GpiQueryCharStringPos

```
EXTRN GpiQueryCharStringPos:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH  DWORD  flOptions    ;Option flag
PUSH  DWORD  lCount       ;Length of the string
PUSH@ OTHER  chString     ;The character string to be examined
PUSH@ OTHER  alXincrements ;Vector of x increment values
PUSH@ OTHER  aptlPositions ;Array of points
CALL GpiQueryCharStringPos

Returns WORD  fSuccess     ;Success indicator
```

GpiQueryCharStringPosAt

```
EXTRN GpiQueryCharStringPosAt:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH@ OTHER  ptlStart     ;Starting position
PUSH  DWORD  flOptions    ;Option flags
PUSH  DWORD  lCount       ;Length of the string
PUSH@ OTHER  chString     ;Character string to be examined
PUSH@ OTHER  alXincrements ;Vector of x increment values
PUSH@ OTHER  aptlPositions ;Array of points, in which the positions of each character in
                           world coordinates are returned
CALL GpiQueryCharStringPosAt

Returns WORD  fSuccess     ;Success indicator
```

GpiQueryClipBox

```
EXTRN GpiQueryClipBox:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH@ OTHER  rc1Bound     ;Bounding rectangle
CALL GpiQueryClipBox

Returns DWORD  lComplexity ;Complexity/error indicator
```

GpiQueryClipRegion

```
EXTRN GpiQueryClipRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
CALL GpiQueryClipRegion

Returns DWORD  hrgn       ;Clip-region handle (if any)
```

GpiQueryColor

```
EXTRN GpiQueryColor:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps          ;Presentation-space handle
CALL GpiQueryColor

Returns DWORD  lColor     ;Color attribute
```

GpiQueryColorData

```
EXTRN GpiQueryColorData:FAR
INCL_GPILOGCOLORTABLE equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Number of elements
PUSH@ OTHER  alArray  ;Array
CALL  GpiQueryColorData

Returns WORD  fSuccess ;Success indicator
```

GpiQueryColorIndex

```
EXTRN GpiQueryColorIndex:FAR
INCL_GPILOGCOLORTABLE equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  flOptions ;Options
PUSH  DWORD  lRgbColor ;Specifies a color in RGB terms
CALL  GpiQueryColorIndex

Returns DWORD lIndex ;Color index providing closest match to the specified color
```

GpiQueryCp

```
EXTRN GpiQueryCp:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
CALL  GpiQueryCp

Returns WORD  usCodePage ;Code page
```

GpiQueryCurrentPosition

```
EXTRN GpiQueryCurrentPosition:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  ptlPoint ;Current position
CALL  GpiQueryCurrentPosition

Returns WORD  fSuccess ;Success indicator
```

GpiQueryDefArcParams

```
EXTRN GpiQueryDefArcParams:FAR
INCL_GPIDEFAULTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  arcpArcParams ;Default arc parameters
CALL  GpiQueryDefArcParams

Returns WORD  fSuccess ;Success indicator
```

GpiQueryDefAttrs

```
EXTRN GpiQueryDefAttrs:FAR
INCL_GPIDEFAULTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lPrimType ;Primitive type
PUSH  DWORD  flAttrMask ;Attributes mask
PUSH@ OTHER  pbunAttrs ;Attributes
CALL  GpiQueryDefAttrs

Returns WORD  fSuccess ;Success indicator
```

GpiQueryDefaultViewMatrix

```
EXTRN GpiQueryDefaultViewMatrix:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Number of elements
PUSH@ OTHER  matlfArray ;Transform matrix
CALL  GpiQueryDefaultViewMatrix

Returns WORD  fSuccess ;Success indicator
```

GpiQueryDefCharBox

```
EXTRN GpiQueryDefCharBox:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  szlSize  ;Default character-box size
CALL  GpiQueryDefCharBox

Returns WORD  fSuccess ;Success indicator
```

GpiQueryDefTag

```
EXTRN GpiQueryDefTag:FAR
INCL_GPIDEFAULTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ DWORD  lTag      ;Default tag identifier
CALL  GpiQueryDefTag

Returns WORD  fSuccess ;Success indicator
```

GpiQueryDefViewingLimits

```
EXTRN GpiQueryDefViewingLimits:FAR
INCL_GPIDEFAULTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  rcLimits ;Default viewing limits
CALL  GpiQueryDefViewingLimits

Returns WORD  fSuccess ;Success indicator
```

GpiQueryDevice

```
EXTRN GpiQueryDevice:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps ;Presentation-space handle
CALL  GpiQueryDevice

Returns DWORD hdc ;Device-context handle
```

GpiQueryDeviceBitmapFormats

```
EXTRN GpiQueryDeviceBitmapFormats:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount   ;Number of elements
PUSH@ OTHER  alArray  ;Data array
CALL  GpiQueryDeviceBitmapFormats

Returns WORD  fSuccess ;Success indicator
```

GpiQueryDrawControl

```
EXTRN GpiQueryDrawControl:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lControl ;Control whose value is to be returned
CALL GpiQueryDrawControl

Returns DWORD lValue ;Value of the control
```

GpiQueryDrawingMode

```
EXTRN GpiQueryDrawingMode:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryDrawingMode

Returns DWORD lMode ;Drawing mode
```

GpiQueryEditMode

```
EXTRN GpiQueryEditMode:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryEditMode

Returns DWORD lMode ;Current editing mode
```

GpiQueryElement

```
EXTRN GpiQueryElement:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lOff ;Starting byte offset within the element
PUSH DWORD lMaxLength ;Maximum length of data that can be returned
PUSH@ OTHER bData ;Element content data
CALL GpiQueryElement

Returns DWORD lRetLength ;Number of bytes returned
```

GpiQueryElementPointer

```
EXTRN GpiQueryElementPointer:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryElementPointer

Returns DWORD lElement ;Current element pointer
```

GpiQueryElementType

```
EXTRN GpiQueryElementType:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ DWORD lType ;Element type
PUSH DWORD lLength ;Data length
PUSH@ ASCIIZ szData ;Description of data buffer
CALL GpiQueryElementType

Returns DWORD lReqLength ;Size of the data required to hold the element content
```


GpiQueryFontFileDescriptions

```
EXTRN GpiQueryFontFileDescriptions:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCIIZ szFilename ;Fully qualified filename
PUSH@ DWORD lCount ;Maximum number of family and facename pairs to be returned
PUSH@ OTHER affdescsNames ;Array of font file descriptors
CALL GpiQueryFontFileDescriptions

Returns DWORD lRemFonts ;Returns
```

GpiQueryFontMetrics

```
EXTRN GpiQueryFontMetrics:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lMetricsLength ;Length of metrics
PUSH@ OTHER fmMetrics ;Metrics of font
CALL GpiQueryFontMetrics

Returns WORD fSuccess ;Success indicator
```

GpiQueryFonts

```
EXTRN GpiQueryFonts:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD flOptions ;Enumeration options
PUSH@ ASCIIZ szFacename ;Facename of fonts
PUSH@ DWORD lReqFonts ;Count of fonts
PUSH DWORD lMetricsLength ;Length of metrics
PUSH@ OTHER afmMetrics ;Metrics of font
CALL GpiQueryFonts

Returns DWORD lRemFonts ;Count of fonts not returned
```

GpiQueryGraphicsField

```
EXTRN GpiQueryGraphicsField:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rclField ;Graphics field
CALL GpiQueryGraphicsField

Returns WORD fSuccess ;Success indicator
```

GpiQueryInitialSegmentAttrs

```
EXTRN GpiQueryInitialSegmentAttrs:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lAttribute ;Attribute to be queried
CALL GpiQueryInitialSegmentAttrs

Returns DWORD lValue ;Current initial attribute value
```

GpiQueryKerningPairs

```
EXTRN GpiQueryKerningPairs:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lCount ;The number of elements in Data
PUSH@ OTHER akrnprData ;Kerning pairs
CALL GpiQueryKerningPairs

Returns DWORD lReturned ;Number returned/error indicator
```

GpiQueryLineEnd

```
EXTRN GpiQueryLineEnd:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryLineEnd

Returns DWORD lLineEnd ;Line end
```

GpiQueryLineJoin

```
EXTRN GpiQueryLineJoin:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryLineJoin

Returns DWORD lLineJoin ;Line join
```

GpiQueryLineType

```
EXTRN GpiQueryLineType:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryLineType

Returns DWORD lLineType ;Line type
```

GpiQueryLineWidth

```
EXTRN GpiQueryLineWidth:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryLineWidth

Returns DWORD fxLineWidth ;Line width
```

GpiQueryLineWidthGeom

```
EXTRN GpiQueryLineWidthGeom:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryLineWidthGeom

Returns DWORD lLineWidth ;Geometric line width
```

GpiQueryLogColorTable

```
EXTRN GpiQueryLogColorTable:FAR
INCL_GPILOGCOLORTABLE equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD fOptions ;Specifies options
PUSH DWORD lStart ;The starting index for which data is to be returned
PUSH DWORD lCount ;Count of elements
PUSH@ OTHER alArray ;An array in which the information is returned
CALL GpiQueryLogColorTable

Returns DWORD lRetCount ;Number of elements returned/error indicator
```

GpiQueryMarker

```
EXTRN GpiQueryMarker:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryMarker

Returns DWORD lSymbol ;Marker symbol
```

GpiQueryMarkerBox

```
EXTRN GpiQueryMarkerBox:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER sizfxSize ;Size of marker box
CALL GpiQueryMarkerBox

Returns WORD fSuccess ;Success indicator
```

GpiQueryMarkerSet

```
EXTRN GpiQueryMarkerSet:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryMarkerSet

Returns DWORD lSet ;Marker set local identifier
```

GpiQueryMetaFileBits

```
EXTRN GpiQueryMetaFileBits:FAR
INCL_GPIMETAFILES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hmf ;Memory-metafile handle
PUSH DWORD lOffset ;Byte offset
PUSH DWORD lLength ;Length in bytes of the metafile data to copy
PUSH@ OTHER bData ;Metafile data
CALL GpiQueryMetaFileBits

Returns WORD fSuccess ;Success indicator
```

GpiQueryMetaFileLength

```
EXTRN GpiQueryMetaFileLength:FAR
INCL_GPIMETAFILES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hmf ;Memory-metafile handle
CALL GpiQueryMetaFileLength

Returns DWORD lLength ;Total length of the metafile
```

GpiQueryMix

```
EXTRN GpiQueryMix:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryMix

Returns DWORD lMixMode ;Mix mode
```

GpiQueryModelTransformMatrix

```
EXTRN GpiQueryModelTransformMatrix:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lCount ;Number of elements
PUSH@ OTHER matlfArray ;Transform matrix
CALL GpiQueryModelTransformMatrix

Returns WORD fSuccess ;Success indicator
```

GpiQueryNearestColor

```
EXTRN GpiQueryNearestColor:FAR
INCL_GPILOGCOLORTABLE equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD flOptions ;Options
PUSH DWORD lRgbIn ;Required color
CALL GpiQueryNearestColor

Returns DWORD lRgbOut ;Nearest available color to the one specified
```

GpiQueryNumberSetIds

```
EXTRN GpiQueryNumberSetIds:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryNumberSetIds

Returns DWORD lCount ;Number of lcids
```

GpiQueryPageViewport

```
EXTRN GpiQueryPageViewport:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rclViewport ;Page viewport
CALL GpiQueryPageViewport

Returns WORD fSuccess ;Success indicator
```

GpiQueryPattern

```
EXTRN GpiQueryPattern:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
CALL GpiQueryPattern

Returns DWORD lPatternSymbol ;Pattern symbol
```

GpiQueryPatternRefPoint

```
EXTRN GpiQueryPatternRefPoint:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps ;Presentation-space handle
PUSH@ OTHER  ptlRefPoint ;Pattern reference point
CALL  GpiQueryPatternRefPoint

Returns WORD  fSuccess ;Success indicator
```

GpiQueryPatternSet

```
EXTRN GpiQueryPatternSet:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps ;Presentation-space handle
CALL  GpiQueryPatternSet

Returns DWORD lSet ;Pattern-set local identifier
```

GpiQueryPel

```
EXTRN GpiQueryPel:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps ;Presentation-space handle
PUSH@ OTHER  ptlPoint ;Position in world coordinates
CALL  GpiQueryPel

Returns DWORD lColor ;Color index of the pel
```

GpiQueryPickAperturePosition

```
EXTRN GpiQueryPickAperturePosition:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps ;Presentation-space handle
PUSH@ OTHER  ptlPoint ;Pick aperture position
CALL  GpiQueryPickAperturePosition

Returns WORD  fSuccess ;Success indicator
```

GpiQueryPickApertureSize

```
EXTRN GpiQueryPickApertureSize:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps ;Presentation-space handle
PUSH@ OTHER  sizlSize ;Pick aperture size
CALL  GpiQueryPickApertureSize

Returns WORD  fSuccess ;Success indicator
```

GpiQueryPS

```
EXTRN GpiQueryPS:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps ;Presentation-space handle
PUSH@ OTHER  sizlSize ;Presentation-page size
CALL  GpiQueryPS

Returns DWORD flOptions ;Presentation-space options
```

GpiQueryRealColors

```
EXTRN GpiQueryRealColors:FAR
INCL_GPILOGCOLORTABLE equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD fOptions ;Options
PUSH DWORD lStart ;Ordinal number of the first color required
PUSH DWORD lCount ;Maximum number of elements
PUSH@ OTHER alColors ;Array in which the information is returned
CALL GpiQueryRealColors

Returns DWORD lRetCount ;Number of elements returned
```

GpiQueryRegionBox

```
EXTRN GpiQueryRegionBox:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD hrgn ;Region handle
PUSH@ OTHER rclBound ;Bounding rectangle
CALL GpiQueryRegionBox

Returns DWORD lComplexity ;Complexity of region/error indicator
```

GpiQueryRegionRects

```
EXTRN GpiQueryRegionRects:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD hrgn ;Region handle
PUSH@ OTHER rclBound ;Bounding rectangle
PUSH@ OTHER rgnrcControl ;Processing-control structure
PUSH@ OTHER arclRects ;Array of rectangle structures, in which the rectangles are
                        returned

CALL GpiQueryRegionRects

Returns WORD fSuccess ;Success indicator
```

GpiQueryRGBColor

```
EXTRN GpiQueryRGBColor:FAR
INCL_GPILOGCOLORTABLE equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD fOptions ;Options
PUSH DWORD lColorIndex ;Color index
CALL GpiQueryRGBColor

Returns DWORD lRgbColor ;RGB color providing closest match to the specified color index
```

GpiQuerySegmentAttrs

```
EXTRN GpiQuerySegmentAttrs:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lSegid ;Segment identifier; must be greater than 0
PUSH DWORD lAttribute ;Attribute to be queried
CALL GpiQuerySegmentAttrs

Returns DWORD lValue ;Current attribute value
```

GpiQuerySegmentNames

```
EXTRN GpiQuerySegmentNames:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lFirstSegid ;First segment in the range; must be greater than 0
PUSH  DWORD  lLastSegid ;Last segment in the range; must be greater than 0
PUSH  DWORD  lMax      ;Maximum number
PUSH@ OTHER  alSegids   ;Array in which the required identifiers are returned
CALL  GpiQuerySegmentNames

Returns DWORD  lRetCount ;Number of identifiers returned
```

GpiQuerySegmentPriority

```
EXTRN GpiQuerySegmentPriority:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lRefSegid ;Reference segment identifier
PUSH  DWORD  lOrder    ;Segment higher or lower
CALL  GpiQuerySegmentPriority

Returns DWORD lSegid    ;Segment identifier
```

GpiQuerySegmentTransformMatrix

```
EXTRN GpiQuerySegmentTransformMatrix:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lSegid    ;Segment identifier
PUSH  DWORD  lCount    ;Number of elements
PUSH@ OTHER  matlfArray ;Transform matrix
CALL  GpiQuerySegmentTransformMatrix

Returns WORD  fSuccess  ;Success indicator
```

GpiQuerySetIds

```
EXTRN GpiQuerySetIds:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount    ;The number of objects to be queried
PUSH@ OTHER  alTypes   ;Object types
PUSH@ OTHER  aNames    ;Font names
PUSH@ OTHER  allcids   ;Local identifiers
CALL  GpiQuerySetIds

Returns WORD  fSuccess  ;Success indicator
```

GpiQueryStopDraw

```
EXTRN GpiQueryStopDraw:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
CALL  GpiQueryStopDraw

Returns DWORD lValue ;Stop draw condition indicator
```

GpiQueryTag

```
EXTRN GpiQueryTag:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ DWORD  lTag     ;Tag identifier
CALL  GpiQueryTag

Returns WORD  fSuccess ;Success indicator
```

GpiQueryTextBox

```
EXTRN GpiQueryTextBox:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount1   ;Number of characters
PUSH@ OTHER  chString  ;The character string
PUSH  DWORD  lCount2   ;Number of points
PUSH@ OTHER  aptlPoints ;List of points
CALL  GpiQueryTextBox

Returns WORD  fSuccess ;Success indicator
```

GpiQueryViewingLimits

```
EXTRN GpiQueryViewingLimits:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  rclLimits ;Viewing limits
CALL  GpiQueryViewingLimits

Returns WORD  fSuccess ;Success indicator
```

GpiQueryViewingTransformMatrix

```
EXTRN GpiQueryViewingTransformMatrix:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lCount    ;Number of elements
PUSH@ OTHER  matlfArray ;Transform matrix
CALL  GpiQueryViewingTransformMatrix

Returns WORD  fSuccess ;Success indicator
```

GpiQueryWidthTable

```
EXTRN GpiQueryWidthTable:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lFirstChar ;Codepoint of first character
PUSH  DWORD  lCount    ;Count of elements in Data
PUSH@ OTHER  alData    ;Array of width values
CALL  GpiQueryWidthTable

Returns WORD  fSuccess ;Success indicator
```

GpiRealizeColorTable

```
EXTRN GpiRealizeColorTable:FAR
INCL_GPILOGCOLORTABLE equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
CALL  GpiRealizeColorTable

Returns WORD  fSuccess ;Success indicator
```


GpiRectInRegion

```
EXTRN GpiRectInRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  hrgn     ;Region handle
PUSH@  OTHER  rcRect   ;Test rectangle
CALL  GpiRectInRegion

Returns DWORD  lInside ;Inside/error indicator
```

GpiRectVisible

```
EXTRN GpiRectVisible:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@  OTHER  rcRectangle ;Test rectangle, in world coordinates
CALL  GpiRectVisible

Returns DWORD  lVisibility ;Visibility indicator
```

GpiRemoveDynamics

```
EXTRN GpiRemoveDynamics:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lFirstSegid ;First segment in the section
PUSH  DWORD  lLastSegid ;Last segment in the section
CALL  GpiRemoveDynamics

Returns WORD  fSuccess ;Success indicator
```

GpiResetBoundaryData

```
EXTRN GpiResetBoundaryData:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
CALL  GpiResetBoundaryData

Returns WORD  fSuccess ;Success indicator
```

GpiResetPS

```
EXTRN GpiResetPS:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  fOptions ;Reset option
CALL  GpiResetPS

Returns WORD  fSuccess ;Success indicator
```

GpiRestorePS

```
EXTRN GpiRestorePS:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lPSid    ;Identifier of the saved presentation space that is to be restored
CALL  GpiRestorePS

Returns WORD  fSuccess ;Success indicator
```

GpiRotate

```
EXTRN GpiRotate:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH@ OTHER  matlfArray    ;Transform matrix
PUSH  DWORD  lOptions     ;Transform options
PUSH  DWORD  fxAngle      ;Rotation angle
PUSH@ OTHER  ptlCenter     ;Center of rotation
CALL  GpiRotate

Returns WORD  fSuccess    ;Success indicator
```

GpiSaveMetaFile

```
EXTRN GpiSaveMetaFile:FAR
INCL_GPIMETAFILES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hmf          ;Metafile handle
PUSH@ ASCIIZ  szFilename  ;Filename
CALL  GpiSaveMetaFile

Returns WORD  fSuccess    ;Success indicator
```

GpiSavePS

```
EXTRN GpiSavePS:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps          ;Presentation-space handle
CALL  GpiSavePS

Returns DWORD lPSid      ;Identifier of saved presentation space
```

GpiScale

```
EXTRN GpiScale:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH@ OTHER  matlfArray    ;Transform matrix
PUSH  DWORD  lOptions     ;Transform options
PUSH@ OTHER  afxScale      ;Scale factors
PUSH@ OTHER  ptlCenter     ;Center of scale
CALL  GpiScale

Returns WORD  fSuccess    ;Success indicator
```

GpiSetArcParams

```
EXTRN GpiSetArcParams:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH@ OTHER  arcpArcParams ;Arc parameters
CALL  GpiSetArcParams

Returns WORD  fSuccess    ;Success indicator
```

GpiSetAttrMode

```
EXTRN GpiSetAttrMode:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH  DWORD  lMode        ;Attribute mode
CALL  GpiSetAttrMode

Returns WORD  fSuccess    ;Success indicator
```

GpiSetAttrs

```
EXTRN GpiSetAttrs:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lPrimType ;Primitive type
PUSH  DWORD  flAttrMask ;Attributes mask
PUSH  DWORD  flDefMask ;Defaults mask
PUSH@  OTHER  pbunAttrs ;Attributes
CALL  GpiSetAttrs

Returns WORD  fSuccess ;Success indicator
```

GpiSetBackColor

```
EXTRN GpiSetBackColor:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lColor    ;Background color
CALL  GpiSetBackColor

Returns WORD  fSuccess ;Success indicator
```

GpiSetBackMix

```
EXTRN GpiSetBackMix:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lMixMode  ;Background-mix mode
CALL  GpiSetBackMix

Returns WORD  fSuccess ;Success indicator
```

GpiSetBitmap

```
EXTRN GpiSetBitmap:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  hbm      ;Handle of the bit map to be set
CALL  GpiSetBitmap

Returns DWORD hbmOld ;Old bit-map handle
```

GpiSetBitmapBits

```
EXTRN GpiSetBitmapBits:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lScanStart ;Line number
PUSH  DWORD  lScans     ;Number of scan lines to be transmitted
PUSH@  OTHER  pBuffer   ;Bit-map data buffer
PUSH@  OTHER  bmiInfoTable ;Bit-map information table
CALL  GpiSetBitmapBits

Returns DWORD lScansSet ;Number of scan lines actually set
```

GpiSetBitmapDimension

```
EXTRN GpiSetBitmapDimension:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hbm      ;Bit-map handle
PUSH@  OTHER  sizlBitmapDimension ;Width and height of bit map
CALL  GpiSetBitmapDimension

Returns WORD  fSuccess ;Success indicator
```

GpiSetBitmapId

```
EXTRN GpiSetBitmapId:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD hbm ;Bit-map handle
PUSH DWORD lLcid ;Local identifier with which the bit map is to be tagged
CALL GpiSetBitmapId

Returns WORD fSuccess ;Success indicator
```

GpiSetCharAngle

```
EXTRN GpiSetCharAngle:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER gradlAngle ;Baseline angle
CALL GpiSetCharAngle

Returns WORD fSuccess ;Success indicator
```

GpiSetCharBox

```
EXTRN GpiSetCharBox:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER sizfBox ;Character-box size in world coordinates
CALL GpiSetCharBox

Returns WORD fSuccess ;Success indicator
```

GpiSetCharDirection

```
EXTRN GpiSetCharDirection:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lDirection ;Character direction
CALL GpiSetCharDirection

Returns WORD fSuccess ;Success indicator
```

GpiSetCharMode

```
EXTRN GpiSetCharMode:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lMode ;Character mode
CALL GpiSetCharMode

Returns WORD fSuccess ;Success indicator
```

GpiSetCharSet

```
EXTRN GpiSetCharSet:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lLcid ;Character-set local identifier
CALL GpiSetCharSet

Returns WORD fSuccess ;Success indicator
```

GpiSetCharShear

```
EXTRN GpiSetCharShear:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  ptlAngle ;Character shear vector
CALL  GpiSetCharShear

Returns WORD  fSuccess ;Success indicator
```

GpiSetClipPath

```
EXTRN GpiSetClipPath:FAR
INCL_GPIPATHS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lPath     ;Path control flag
PUSH  DWORD  lOptions  ;Options
CALL  GpiSetClipPath

Returns WORD  fSuccess ;Success indicator
```

GpiSetClipRegion

```
EXTRN GpiSetClipRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  hrgn     ;Region handle
PUSH@ DWORD  hrgnOld   ;Old region handle (if any)
CALL  GpiSetClipRegion

Returns DWORD  lComplexity ;Complexity of clipping/error indicator
```

GpiSetColor

```
EXTRN GpiSetColor:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lColor    ;Color
CALL  GpiSetColor

Returns WORD  fSuccess ;Success indicator
```

GpiSetCp

```
EXTRN GpiSetCp:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  WORD   usCodePage ;Code-page id
CALL  GpiSetCp

Returns WORD  fSuccess ;Success indicator
```

GpiSetCurrentPosition

```
EXTRN GpiSetCurrentPosition:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  ptlPoint ;New value of current position
CALL  GpiSetCurrentPosition

Returns WORD  fSuccess ;Success indicator
```

GpiSetDefArcParams

```
EXTRN GpiSetDefArcParams:FAR
INCL_GPIDEFAULTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH@ OTHER  arcpArcParams ;Default arc parameters
CALL  GpiSetDefArcParams

Returns WORD  fSuccess      ;Success indicator
```

GpiSetDefAttrs

```
EXTRN GpiSetDefAttrs:FAR
INCL_GPIDEFAULTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH  DWORD  lPrimType    ;Primitive type
PUSH  DWORD  flAttrMask   ;Attributes mask
PUSH@ OTHER  pbunAttrs    ;Default attribute values
CALL  GpiSetDefAttrs

Returns WORD  fSuccess     ;Success indicator
```

GpiSetDefaultViewMatrix

```
EXTRN GpiSetDefaultViewMatrix:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH  DWORD  lCount       ;Number of elements
PUSH@ OTHER  matlfarray   ;Transformation matrix
PUSH  DWORD  lOptions     ;Transform options
CALL  GpiSetDefaultViewMatrix

Returns WORD  fSuccess     ;Success indicator
```

GpiSetDefTag

```
EXTRN GpiSetDefTag:FAR
INCL_GPIDEFAULTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH  DWORD  lTag         ;Default tag identifier
CALL  GpiSetDefTag

Returns WORD  fSuccess     ;Success indicator
```

GpiSetDefViewingLimits

```
EXTRN GpiSetDefViewingLimits:FAR
INCL_GPIDEFAULTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH@ OTHER  rclLimits    ;Default viewing limits
CALL  GpiSetDefViewingLimits

Returns WORD  fSuccess     ;Success indicator
```

GpiSetDrawControl

```
EXTRN GpiSetDrawControl:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps          ;Presentation-space handle
PUSH  DWORD  lControl     ;Drawing control
PUSH  DWORD  lValue       ;Required value of the drawing control
CALL  GpiSetDrawControl

Returns WORD  fSuccess     ;Success indicator
```

GpiSetDrawingMode

```
EXTRN GpiSetDrawingMode:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lMode ;Mode to be used for subsequent drawing calls
CALL GpiSetDrawingMode

Returns WORD fSuccess ;Success indicator
```

GpiSetEditMode

```
EXTRN GpiSetEditMode:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lMode ;Edit mode
CALL GpiSetEditMode

Returns WORD fSuccess ;Success indicator
```

GpiSetElementPointer

```
EXTRN GpiSetElementPointer:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lElement ;The element number required
CALL GpiSetElementPointer

Returns WORD fSuccess ;Success indicator
```

GpiSetElementPointerAtLabel

```
EXTRN GpiSetElementPointerAtLabel:FAR
INCL_GPISEGEDITING equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lLabel ;Required label
CALL GpiSetElementPointerAtLabel

Returns WORD fSuccess ;Success indicator
```

GpiSetGraphicsField

```
EXTRN GpiSetGraphicsField:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rcIField ;Graphics field
CALL GpiSetGraphicsField

Returns WORD fSuccess ;Success indicator
```

GpiSetInitialSegmentAttrs

```
EXTRN GpiSetInitialSegmentAttrs:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lAttribute ;Segment attribute
PUSH DWORD lValue ;Attribute value
CALL GpiSetInitialSegmentAttrs

Returns WORD fSuccess ;Success indicator
```

GpiSetLineEnd

```
EXTRN GpiSetLineEnd:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lLineEnd ;Style of line end
CALL GpiSetLineEnd

Returns WORD fSuccess ;Success indicator
```

GpiSetLineJoin

```
EXTRN GpiSetLineJoin:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lLineJoin ;Style of line join
CALL GpiSetLineJoin

Returns WORD fSuccess ;Success indicator
```

GpiSetLineType

```
EXTRN GpiSetLineType:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lLineType ;Line types available
CALL GpiSetLineType

Returns WORD fSuccess ;Success indicator
```

GpiSetLineWidth

```
EXTRN GpiSetLineWidth:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD fxLineWidth ;Line-width multiplier
CALL GpiSetLineWidth

Returns WORD fSuccess ;Success indicator
```

GpiSetLineWidthGeom

```
EXTRN GpiSetLineWidthGeom:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lLineWidth ;Geometric line width
CALL GpiSetLineWidthGeom

Returns WORD fSuccess ;Success indicator
```

GpiSetMarker

```
EXTRN GpiSetMarker:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lSymbol ;Marker symbol
CALL GpiSetMarker

Returns WORD fSuccess ;Success indicator
```


GpiSetMarkerBox

```
EXTRN GpiSetMarkerBox:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER sizfxSize ;Size of marker box
CALL GpiSetMarkerBox

Returns WORD fSuccess ;Success indicator
```

GpiSetMarkerSet

```
EXTRN GpiSetMarkerSet:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lSet ;Marker-set local identifier
CALL GpiSetMarkerSet

Returns WORD fSuccess ;Success indicator
```

GpiSetMetaFileBits

```
EXTRN GpiSetMetaFileBits:FAR
INCL_GPIMETAFILES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hmf ;Metafile-memory handle
PUSH DWORD lOffset ;Offset
PUSH DWORD lLength ;Length of the metafile data
PUSH@ OTHER bBuffer ;Metafile data buffer
CALL GpiSetMetaFileBits

Returns WORD fSuccess ;Success indicator
```

GpiSetMix

```
EXTRN GpiSetMix:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lMixMode ;Mix mode
CALL GpiSetMix

Returns WORD fSuccess ;Success indicator
```

GpiSetModelTransformMatrix

```
EXTRN GpiSetModelTransformMatrix:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lCount ;Number of elements in matrix
PUSH@ OTHER matlfArray ;Transformation matrix
PUSH DWORD lOptions ;Transform options
CALL GpiSetModelTransformMatrix

Returns WORD fSuccess ;Success indicator
```

GpiSetPageViewport

```
EXTRN GpiSetPageViewport:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rcViewport ;Page viewport
CALL GpiSetPageViewport

Returns WORD fSuccess ;Success indicator
```

GpiSetPattern

```
EXTRN GpiSetPattern:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER 1PatternSymbol ;Pattern symbol
CALL GpiSetPattern

Returns WORD fSuccess ;Success indicator
```

GpiSetPatternRefPoint

```
EXTRN GpiSetPatternRefPoint:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptlRefPoint ;Pattern reference point
CALL GpiSetPatternRefPoint

Returns WORD fSuccess ;Success indicator
```

GpiSetPatternSet

```
EXTRN GpiSetPatternSet:FAR
INCL_GPIPRIMITIVES equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD 1Set ;Pattern-set local identifier
CALL GpiSetPatternSet

Returns WORD fSuccess ;Success indicator
```

GpiSetPel

```
EXTRN GpiSetPel:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptlPoint ;Position in world coordinates
CALL GpiSetPel

Returns DWORD 1Hits ;Correlation/error indicator
```

GpiSetPickAperturePosition

```
EXTRN GpiSetPickAperturePosition:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER ptlPick ;Center of the pick aperture
CALL GpiSetPickAperturePosition

Returns WORD fSuccess ;Success indicator
```

GpiSetPickApertureSize

```
EXTRN GpiSetPickApertureSize:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD 1Options ;Setting option
PUSH@ OTHER sizlSize ;Pick aperture size
CALL GpiSetPickApertureSize

Returns WORD fSuccess ;Success indicator
```

GpiSetPS

```
EXTRN GpiSetPS:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@ OTHER  sizsize  ;Presentation-space size
PUSH  DWORD  flOptions ;Options
CALL  GpiSetPS

Returns WORD  fSuccess ;Success indicator
```

GpiSetRegion

```
EXTRN GpiSetRegion:FAR
INCL_GPIREGIONS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  hrgn      ;Region handle
PUSH  DWORD  lcount    ;Count of rectangles
PUSH@ OTHER  arclRectangles ;Array of rectangles
CALL  GpiSetRegion

Returns WORD  fSuccess ;Success indicator
```

GpiSetSegmentAttrs

```
EXTRN GpiSetSegmentAttrs:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lSegid    ;Segment identifier
PUSH  DWORD  lAttribute ;Segment attribute
PUSH  DWORD  lValue     ;Attribute value
CALL  GpiSetSegmentAttrs

Returns WORD  fSuccess ;Success indicator
```

GpiSetSegmentPriority

```
EXTRN GpiSetSegmentPriority:FAR
INCL_GPISEGMENTS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lSegid    ;Segment identifier
PUSH  DWORD  lRefSegid ;Reference segment identifier
PUSH  DWORD  lOrder     ;Segment higher or lower
CALL  GpiSetSegmentPriority

Returns WORD  fSuccess ;Success indicator
```

GpiSetSegmentTransformMatrix

```
EXTRN GpiSetSegmentTransformMatrix:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  DWORD  lSegid    ;Segment identifier
PUSH  DWORD  lCount     ;Number of elements
PUSH@ OTHER  matlfarray ;Transformation matrix
PUSH  DWORD  lOptions   ;Transform options
CALL  GpiSetSegmentTransformMatrix

Returns WORD  fSuccess ;Success indicator
```

GpiSetStopDraw

```
EXTRN GpiSetStopDraw:FAR
INCL_GPICONTROL equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lValue ;Stop draw condition
CALL GpiSetStopDraw

Returns WORD fSuccess ;Success indicator
```

GpiSetTag

```
EXTRN GpiSetTag:FAR
INCL_GPICORRELATION equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lTag ;Tag identifier
CALL GpiSetTag

Returns WORD fSuccess ;Success indicator
```

GpiSetViewingLimits

```
EXTRN GpiSetViewingLimits:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rcLimits ;Viewing limits in model space
CALL GpiSetViewingLimits

Returns WORD fSuccess ;Success indicator
```

GpiSetViewingTransformMatrix

```
EXTRN GpiSetViewingTransformMatrix:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lCount ;Number of elements
PUSH@ OTHER matlfArray ;Transformation matrix
PUSH DWORD lOptions ;Transform option
CALL GpiSetViewingTransformMatrix

Returns WORD fSuccess ;Success indicator
```

GpiStrokePath

```
EXTRN GpiStrokePath:FAR
INCL_GPIPATHS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH DWORD lPath ;Identifier of path to be stroked; it must be 1
PUSH DWORD flOptions ;Stroke option
CALL GpiStrokePath

Returns DWORD lHits ;Correlation/error indicator
```

GpiTranslate

```
EXTRN GpiTranslate:FAR
INCL_GPITRANSFORMS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER matlfArray ;Transform matrix
PUSH DWORD lOptions ;Transform options
PUSH@ OTHER ptlTranslation ;Translation
CALL GpiTranslate

Returns WORD fSuccess ;Success indicator
```

GpiUnloadFonts

```
EXTRN GpiUnloadFonts:FAR
INCL_GPILCIDS equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCIIZ szFilename ;Fully qualified file name of the font resource
CALL GpiUnloadFonts

Returns WORD fSuccess ;Success indicator
```

GpiUnrealizeColorTable

```
EXTRN GpiUnrealizeColorTable:FAR
INCL_GPILOGCOLORTABLE equ 1 ;Or use INCL_GPI or INCL_PM

PUSH DWORD hps ;Presentation-space handle
CALL GpiUnrealizeColorTable

Returns WORD fSuccess ;Success indicator
```

GpiWCBitBlt

```
EXTRN GpiWCBitBlt:FAR
INCL_GPIBITMAPS equ 1 ;Or use INCL_GPI or INCL_PM. Also in COMMON section

PUSH DWORD hpsTarget ;Target presentation-space handle
PUSH DWORD hbmSource ;Source bit-map handle
PUSH DWORD lCount ;Point count
PUSH@ OTHER aptlPoints ;Point array
PUSH DWORD lRop ;Mixing function required
PUSH DWORD flOptions ;Options
CALL GpiWCBitBlt

Returns DWORD lHits ;Correlation/error indicator
```

Chapter 5. Picture Function Calls

PicIchg

```
EXTRN PicIchg:FAR
INCL_PIC equ 1 ;Or use INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCIIZ szSourceFileName ;The name of the source file
PUSH@ ASCIIZ szDestinationFileName ;The name of the destination file
PUSH DWORD lType ;Specifies the type of conversion
CALL PicIchg

Returns WORD fSuccess ;Success indicator
```

PicPrint

```
EXTRN PicPrint:FAR
INCL_PIC equ 1 ;Or use INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCIIZ szFilename ;Full path and filename of the source file
PUSH DWORD lType ;Type of source file
PUSH@ ASCIIZ szParams ;Spooler parameters
CALL PicPrint

Returns WORD fSuccess ;Success indicator
```

Chapter 6. Profile Function Calls

PrfAddProgram

```
EXTRN PrfAddProgram:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hini ;Initialization-file handle
PUSH@ OTHER progdeDetails ;Program details for the program to be added to the program list
PUSH DWORD hprogGroup ;Handle of the group to which the program is to be added
CALL PrfAddProgram

Returns DWORD hprog ;Program handle for the program added to the program list
```

PrfChangeProgram

```
EXTRN PrfChangeProgram:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hini ;Initialization-file handle
PUSH DWORD hprog ;Handle of the program whose information is to be changed
PUSH@ OTHER Details ;Program details
CALL PrfChangeProgram

Returns WORD fSuccess ;Success indicator
```

PrfCloseProfile

```
EXTRN PrfCloseProfile:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hini ;Initialization-file handle
CALL PrfCloseProfile

Returns WORD fSuccess ;Success indicator
```

PrfCreateGroup

```
EXTRN PrfCreateGroup:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hini ;Initialization-file handle
PUSH@ ASCIIZ szTitle ;Title of the new group
PUSH WORD chVisibility ;Visibility control
CALL PrfCreateGroup

Returns DWORD hprogGroup ;Group handle for the newly-created group
```

PrfDestroyGroup

```
EXTRN PrfDestroyGroup:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hini ;Initialization-file handle
PUSH DWORD hprogGroup ;Group handle
CALL PrfDestroyGroup

Returns WORD fSuccess ;Success indicator
```


PrfOpenProfile

```
EXTRN PrfOpenProfile:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH@ ASCIIZ  szFileName ;User-profile file name
CALL  PrfOpenProfile

Returns DWORD  hini          ;Initialization-file handle
```

PrfQueryDefinition

```
EXTRN PrfQueryDefinition:FAR
INCL_WINPROGAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hini          ;Initialization-file handle
PUSH  DWORD  hprog         ;Handle of the program whose details are to be returned
PUSH@ OTHER  Details       ;Program details
PUSH  DWORD  cchBufferMax  ;Maximum length in bytes
CALL  PrfQueryDefinition

Returns DWORD  ulLength    ;Length of returned data
```

PrfQueryProfile

```
EXTRN PrfQueryProfile:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH@ OTHER  prfproProfile ;Profile names structure
CALL  PrfQueryProfile

Returns WORD   fSuccess    ;Success indicator
```

PrfQueryProfileData

```
EXTRN PrfQueryProfileData:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hini          ;Initialization-file handle
PUSH@ ASCIIZ  szApp        ;Application name
PUSH@ ASCIIZ  szKey        ;Key name
PUSH@ OTHER  Buffer        ;Value data
PUSH@ DWORD  ulBufferMax   ;Size of value data
CALL  PrfQueryProfileData

Returns WORD   fSuccess    ;Success indicator
```

PrfQueryProfileInt

```
EXTRN PrfQueryProfileInt:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hini          ;Initialization-file handle
PUSH@ ASCIIZ  szApp        ;Application name
PUSH@ ASCIIZ  szKey        ;Key name
PUSH  WORD   sDefault      ;Default value
CALL  PrfQueryProfileInt

Returns WORD   sResult     ;Key value specified in the initialization file
```

PrfQueryProfileSize

```
EXTRN PrfQueryProfileSize:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hini      ;Initialization-file handle
PUSH@ ASCIIZ  szApp     ;Application name
PUSH@ ASCIIZ  szKey     ;Key name
PUSH@ DWORD   DataLen   ;Data length
CALL PrfQueryProfileSize

Returns WORD  fSuccess ;Success indicator
```

PrfQueryProfileString

```
EXTRN PrfQueryProfileString:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hini      ;Initialization-file handle
PUSH@ ASCIIZ  szApp     ;Application name
PUSH@ ASCIIZ  szKey     ;Key name
PUSH@ ASCIIZ  szDefault ;Default string
PUSH@ OTHER   Buffer     ;Profile string
PUSH  DWORD   cchBufferMax ;Maximum string length
CALL PrfQueryProfileString

Returns DWORD  pulLength ;String length returned
```

PrfQueryProgramCategory

```
EXTRN PrfQueryProgramCategory:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hini      ;Initialization-file handle
PUSH@ ASCIIZ  szExe     ;Executable-file name
CALL PrfQueryProgramCategory

Returns WORD   Category ;Program category
```

PrfQueryProgramHandle

```
EXTRN PrfQueryProgramHandle:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hini      ;Initialization-file handle
PUSH@ ASCIIZ  szExe     ;Executable-file name
PUSH@ OTHER   hprogArray ;Program handles
PUSH  DWORD   cchBufferMax ;Maximum number of program handles
PUSH@ DWORD   ulCount    ;Number of program handles returned
CALL PrfQueryProgramHandle

Returns DWORD  ulLength ;Number of bytes returned
```

PrfQueryProgramTitles

```
EXTRN PrfQueryProgramTitles:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hini      ;Initialization-file handle
PUSH  DWORD  hprogGroup ;Handle of the program or group whose information is to be
                        returned
PUSH@ OTHER   Titles     ;Program information buffer
PUSH  DWORD   cchBufferMax ;Buffer length
PUSH@ DWORD   ulCount    ;Number of structures returned
CALL PrfQueryProgramTitles

Returns DWORD  ulLength ;Length returned
```

PrfRemoveProgram

```
EXTRN PrfRemoveProgram:FAR
INCL_WINPROGRLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hini ;Initialization-file handle
PUSH DWORD hprog ;Program handle
CALL PrfRemoveProgram

Returns WORD fSuccess ;Success indicator
```

PrfReset

```
EXTRN PrfReset:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER prfproProfile ;Profile-names structure
CALL PrfReset

Returns WORD fSuccess ;Success indicator
```

PrfWriteProfileData

```
EXTRN PrfWriteProfileData:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hini ;Initialization-file handle
PUSH@ ASCIIZ szApp ;Application name
PUSH@ ASCIIZ szKey ;Key name
PUSH@ OTHER Data ;Value data
PUSH DWORD cchDataLen ;Size of value data
CALL PrfWriteProfileData

Returns WORD fSuccess ;Success indicator
```

PrfWriteProfileString

```
EXTRN PrfWriteProfileString:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hini ;Initialization-file handle
PUSH@ ASCIIZ szApp ;Application name
PUSH@ ASCIIZ szKey ;Key name
PUSH@ ASCIIZ szData ;Text string
CALL PrfWriteProfileString

Returns WORD fSuccess ;Success indicator
```

Chapter 7. Spooler Function Calls

SplQmAbort

```
EXTRN SplQmAbort:FAR
INCL_SPL equ 1 ;Or use INCL_PM

PUSH DWORD hsp1 ;Spooler handle
CALL SplQmAbort

Returns WORD fSuccess ;Success indicator
```

SplQmClose

```
EXTRN SplQmClose:FAR
INCL_SPL equ 1 ;Or use INCL_PM

PUSH DWORD hsp1 ;Spooler handle
CALL SplQmClose

Returns WORD fSuccess ;Success indicator
```

SplQmEndDoc

```
EXTRN SplQmEndDoc:FAR
INCL_SPL equ 1 ;Or use INCL_PM

PUSH DWORD hsp1 ;Spooler handle
CALL SplQmEndDoc

Returns WORD usJob ;Job identifier
```

SplQmOpen

```
EXTRN SplQmOpen:FAR
INCL_SPL equ 1 ;Or use INCL_PM

PUSH@ ASCIIZ szToken ;A token (nickname) that identifies spooler information
PUSH DWORD lCount ;Number of items
PUSH@ OTHER qmdopData ;Open parameters
CALL SplQmOpen

Returns DWORD hsp1 ;Spooler handle
```

SplQmStartDoc

```
EXTRN SplQmStartDoc:FAR
INCL_SPL equ 1 ;Or use INCL_PM

PUSH DWORD hsp1 ;Spooler handle
PUSH@ ASCIIZ szDocName ;Document name
CALL SplQmStartDoc

Returns WORD fSuccess ;Success indicator
```

SplQmWrite

```
EXTRN SplQmWrite:FAR
INCL_SPL equ 1 ;Or use INCL_PM

PUSH DWORD hsp1 ;Spooler handle
PUSH DWORD lCount ;Length in bytes
PUSH@ OTHER bData ;Buffer of data to be written to the spool file
CALL SplQmWrite

Returns WORD fSuccess ;Success indicator
```

SplQpInstall

```
EXTRN SplQpInstall:FAR
INCL_SPL equ 1 ;Or use INCL_PM

PUSH DWORD hwnd ;Window handle
CALL SplQpInstall

Returns WORD fSuccess ;Success indicator
```

SplQpQueryDt

```
EXTRN SplQpQueryDt:FAR
INCL_SPL equ 1 ;Or use INCL_PM

PUSH@ DWORD lCount ;Maximum number of data types that can be returned
PUSH@ ASCIIZ aszDatatypes ;An array containing the data types supported
CALL SplQpQueryDt

Returns WORD fSuccess ;Success indicator
```

Chapter 8. Video Function Calls

VioAssociate

```
EXTRN VioAssociate:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH DWORD hdc ;Device-context handle
PUSH WORD hvps ;VIO presentation-space handle
CALL VioAssociate

Returns WORD usError ;Error indicator
```

VioCreateLogFont

```
EXTRN VioCreateLogFont:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH@ OTHER fatatrs ;Attributes of font
PUSH DWORD llcid ;Local identifier
PUSH@ OTHER Name ;Logical-font name
PUSH WORD hvps ;VIO presentation-space handle
CALL VioCreateLogFont

Returns WORD usError ;Error indicator
```

VioCreatePS

```
EXTRN VioCreatePS:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH@ WORD hvps ;VIO presentation-space handle
PUSH WORD sdepth ;Presentation-space depth
PUSH WORD swidth ;Presentation-space width
PUSH WORD sFormat ;Presentation-space format
PUSH WORD sAttrs ;Attribute bytes per character
PUSH WORD hvpsReserved ;Reserved (must be zero)
CALL VioCreatePS

Returns WORD usError ;Error indicator
```

VioDeleteSetId

```
EXTRN VioDeleteSetId:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH DWORD llcid ;Local identifier
PUSH WORD hvps ;VIO presentation-space handle
CALL VioDeleteSetId

Returns WORD usError ;Error indicator
```

VioDestroyPS

```
EXTRN VioDestroyPS:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH WORD hvps ;VIO presentation-space handle
CALL VioDestroyPS

Returns WORD usError ;Error indicator
```

VioGetDeviceCellSize

```
EXTRN VioGetDeviceCellSize:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH@ WORD sHeight ;Current device cell height in pels
PUSH@ WORD sWidth ;Current device cell width in pels
PUSH WORD hvps ;VIO presentation-space handle
CALL VioGetDeviceCellSize

Returns WORD usError ;Error indicator
```

VioGetOrg

```
EXTRN VioGetOrg:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH@ WORD sRow ;Row number
PUSH@ WORD sColumn ;Column number
PUSH WORD hvps ;VIO presentation-space handle
CALL VioGetOrg

Returns WORD usError ;Error indicator
```

VioQueryFonts

```
EXTRN VioQueryFonts:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH@ DWORD lRemfonts ;Number of fonts for which information is not returned
PUSH@ OTHER afmMetrics ;Font metrics
PUSH@ DWORD lMetricsLength ;The length of each metrics record to be returned
PUSH@ DWORD lFonts ;Number of fonts
PUSH@ ASCIIZ szFacename ;Facename of the fonts of interest
PUSH@ DWORD flOptions ;Enumeration options
PUSH WORD hvps ;VIO presentation-space handle
CALL VioQueryFonts

Returns WORD usError ;Error indicator
```

VioQuerySetIds

```
EXTRN VioQuerySetIds:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH@ OTHER allcids ;Local identifiers
PUSH@ OTHER Names ;Font names
PUSH@ OTHER alTypes ;Object types
PUSH@ DWORD lcount ;The number of objects to be queried
PUSH WORD hvps ;VIO presentation-space handle
CALL VioQuerySetIds

Returns WORD usError ;Error indicator
```

VioSetDeviceCellSize

```
EXTRN VioSetDeviceCellSize:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH WORD sHeight ;The required device cell height in pels
PUSH WORD sWidth ;The required device cell width in pels
PUSH WORD hvps ;VIO presentation space handle
CALL VioSetDeviceCellSize

Returns WORD usError ;Error indicator
```

VioSetOrg

```
EXTRN VioSetOrg:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH WORD sRow ;Row number
PUSH WORD sColumn ;Column number
PUSH WORD hvps ;VIO presentation-space handle
CALL VioSetOrg

Returns WORD usError ;Error indicator
```

VioShowPS

```
EXTRN VioShowPS:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH WORD sDepth ;The depth of the rectangle in character cell units
PUSH WORD sWidth ;The width of the rectangle in character cell units
PUSH WORD soffCell ;Cell offset
PUSH WORD hvps ;VIO presentation-space handle
CALL VioShowPS

Returns WORD usError ;Error indicator
```

Chapter 9. Window Function Calls

WinAddAtom

```
EXTRN WinAddAtom:FAR
INCL_WINATOM equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hatomtblAtomTbl ;Atom-table handle
PUSH@ ASCIIZ szAtomName ;Atom name
CALL WinAddAtom

Returns WORD atom ;Atom value
```

WinAddProgram

```
EXTRN WinAddProgram:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER pibProgramInfo ;Program information for the program to be added to the
                                installed program list
PUSH DWORD hprogGroupHandle ;Handle of the program group to which the program is to be
                                added

CALL WinAddProgram

Returns DWORD hprogProgHandle ;Program handle for the program added to the installed
                                program list
```

WinAddSwitchEntry

```
EXTRN WinAddSwitchEntry:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH@ OTHER swctlSwitchData ;Switch data
CALL WinAddSwitchEntry

Returns DWORD hswitchSwitch ;Handle to the newly-created switch list entry
```

WinAlarm

```
EXTRN WinAlarm:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwndDeskTop ;Desktop-window handle
PUSH WORD fsStyle ;Alarm style
CALL WinAlarm

Returns WORD fResult ;Alarm-generated indicator
```

WinAllocMem

```
EXTRN WinAllocMem:FAR
INCL_WINHEAP equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hheapHeap ;Heap handle
PUSH WORD uscb ;Total number of bytes to allocate
CALL WinAllocMem

Returns WORD npbMem ;Memory pointer
```

WinAvailMem

```
EXTRN WinAvailMem:FAR
INCL_WINHEAP equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hheapHeap ;Heap handle
PUSH WORD fCompact ;Compact indicator
PUSH WORD cb ;Reserved
CALL WinAvailMem

Returns WORD usLargest ;Size of the largest memory block available
```

WinBeginEnumWindows

```
EXTRN WinBeginEnumWindows:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndParent ;Handle of the window whose child windows are to be enumerated
CALL WinBeginEnumWindows

Returns DWORD Henum ;Enumeration handle
```

WinBeginPaint

```
EXTRN WinBeginPaint:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Handle of window where drawing is going to occur
PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rcRect ;Bounding rectangle
CALL WinBeginPaint

Returns DWORD hpsPaintPS ;Presentation-space handle
```

WinBroadcastMsg

```
EXTRN WinBroadcastMsg:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndParent ;Parent-window handle
PUSH WORD usMsgId ;Message identifier
PUSH DWORD mpParam1 ;Parameter 1
PUSH DWORD mpParam2 ;Parameter 2
PUSH WORD fsCmd ;Broadcast message command
CALL WinBroadcastMsg

Returns WORD fSuccess ;Success indicator
```

WinCalcFrameRect

```
EXTRN WinCalcFrameRect:FAR
INCL_WINFRAMEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Frame-window handle
PUSH@ OTHER rcRect ;Window rectangle
PUSH WORD fFrame ;Frame indicator
CALL WinCalcFrameRect

Returns WORD fSuccess ;Rectangle-calculated indicator
```

WinCallMsgFilter

```
EXTRN WinCallMsgFilter:FAR
INCL_WINHOOKS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER qmsgpqmsg ;Message to be passed to the message-filter hook
PUSH WORD usFilter ;Filter
CALL WinCallMsgFilter

Returns WORD fHookRet ;Message-filter hook return indicator
```

WinCancelShutdown

```
EXTRN WinCancelShutdown:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hmq ;Queue handle
PUSH WORD fCancelAlways ;Cancellation control
CALL WinCancelShutdown

Returns WORD fSuccess ;Success indicator
```

WinCatch

```
EXTRN WinCatch:FAR
INCL_WINCATCHTHROW equ 1 ;Or use INCL_WIN or INCL_PM

PUSH@ OTHER ctchbfCatchBuf ;Saved execution environment buffer
CALL WinCatch

Returns WORD sRetCode ;Return code
```

WinChangeSwitchEntry

```
EXTRN WinChangeSwitchEntry:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hswitchSwitch ;Handle to the switch-list entry to be changed
PUSH@ OTHER swctlSwitchData ;Switch-control data
CALL WinChangeSwitchEntry

Returns WORD usRetCode ;Return code
```

WinCloseClipbrd

```
EXTRN WinCloseClipbrd:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
CALL WinCloseClipbrd

Returns WORD fSuccess ;Success indicator
```

WinCompareStrings

```
EXTRN WinCompareStrings:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH WORD idCodepage ;Codepage identity of both strings
PUSH WORD idCountryCode ;Country code
PUSH@ ASCIIZ szString1 ;String 1
PUSH@ ASCIIZ szString2 ;String 2
PUSH WORD fsOptions ;Reserved
CALL WinCompareStrings

Returns WORD usResult ;Comparison result
```

WinCopyAccelTable

```
EXTRN WinCopyAccelTable:FAR
INCL_WINACCELERATORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hAccel ;Accelerator-table handle
PUSH@ OTHER acctAccelTable ;Accelerator-table data area
PUSH WORD usCopyMax ;Maximum data area size
CALL WinCopyAccelTable

Returns WORD usCopied ;Amount copied or size required
```

WinCopyRect

```
EXTRN WinCopyRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  rc1Dest  ;Destination rectangle
PUSH@ OTHER  rc1Src   ;Source rectangle
CALL WinCopyRect

Returns WORD  fSuccess ;Success indicator
```

WinCpTranslateChar

```
EXTRN WinCpTranslateChar:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   idCpSource ;Source-character code page
PUSH  WORD   ucSource  ;Character to be translated
PUSH  WORD   idCpDest  ;Code page of the resultant character
CALL WinCpTranslateChar

Returns WORD  ucDest    ;If nonzero, the translated character
```

WinCpTranslateString

```
EXTRN WinCpTranslateString:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   idCpSource ;Source-string code page
PUSH@ ASCIIZ  szSource  ;String to be translated
PUSH  WORD   idCpDest  ;Code page of the resultant string
PUSH  WORD   cbLenDest ;Maximum length of output string
PUSH@ ASCIIZ  szDest    ;The translated string
CALL WinCpTranslateString

Returns WORD  fSuccess ;Success indicator
```

WinCreateAccelTable

```
EXTRN WinCreateAccelTable:FAR
INCL_WINACCELERATORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  acctAccelTable ;Accelerator table
CALL WinCreateAccelTable

Returns DWORD  hAccel      ;Accelerator-table handle
```

WinCreateAtomTable

```
EXTRN WinCreateAtomTable:FAR
INCL_WINATOM equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  WORD  usInitial      ;Initial bytes
PUSH  WORD  usBuckets      ;Size of the hash table
CALL WinCreateAtomTable

Returns DWORD  hatomtblAtomTbl ;Atom-table handle
```

WinCreateCursor

```
EXTRN WinCreateCursor:FAR
INCL_WINCURSORS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Handle of window in which cursor is displayed
PUSH WORD sx ;x position of cursor
PUSH WORD sy ;y position of cursor
PUSH WORD scx ;x size of cursor
PUSH WORD scy ;y size of cursor
PUSH WORD usrgf ;Controls the appearance of the cursor
PUSH@ OTHER rcClip ;Cursor rectangle
CALL WinCreateCursor

Returns WORD fSuccess ;Success indicator
```

WinCreateDlg

```
EXTRN WinCreateDlg:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndParent ;Parent-window handle of the created dialog window
PUSH DWORD hwndOwner ;Requested owner-window handle of the created dialog window
PUSH@ OTHER DlgProc ;Dialog procedure for the created dialog window
PUSH@ OTHER dlgDlgTmp ;Dialog template
PUSH@ OTHER CreateParams ;Application-defined data area
CALL WinCreateDlg

Returns DWORD hwndDlg ;Dialog-window handle
```

WinCreateFrameControls

```
EXTRN WinCreateFrameControls:FAR
INCL_WINFRAMEMEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndFrame ;Frame-window handle
PUSH@ OTHER Fcdata ;Frame-control data
PUSH@ ASCIIZ szTitle ;Title string
CALL WinCreateFrameControls

Returns WORD fSuccess ;Success indicator
```

WinCreateGroup

```
EXTRN WinCreateGroup:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCIIZ szTitle ;Title of the new group
PUSH WORD ucVisibility ;Visibility control
PUSH DWORD flres1 ;Reserved
PUSH DWORD flres2 ;Reserved
CALL WinCreateGroup

Returns DWORD hprogGroupHandle ;Program-group handle for the newly-created group
```

WinCreateHeap

```
EXTRN WinCreateHeap:FAR
INCL_WINHEAP equ 1 ;Or use INCL_WIN or INCL_PM

PUSH WORD usHeapBase ;Selector of the segment to contain the local heap
PUSH WORD usHeapSize ;Initial heap size in bytes
PUSH WORD usGrow ;Heap growth size in bytes
PUSH WORD usMinDed ;Minimum element size
PUSH WORD usMaxDed ;Maximum element size
PUSH WORD fsOptions ;Optional characteristics
CALL WinCreateHeap

Returns DWORD hHeap ;Heap handle
```

WinCreateMenu

```
EXTRN WinCreateMenu:FAR
INCL_WINMENUS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndOwner ;Owner- and parent-window handle of the created menu window
PUSH@ OTHER mtMenuTpm ;Menu template in binary format
CALL WinCreateMenu

Returns DWORD hwndMenu ;Menu-window handle
```

WinCreateMsgQueue

```
EXTRN WinCreateMsgQueue:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
PUSH WORD sQueueSize ;Maximum queue size
CALL WinCreateMsgQueue

Returns DWORD hmq ;Message-queue handle
```

WinCreatePointer

```
EXTRN WinCreatePointer:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndDesktop ;Desktop-window handle or HWND_DESKTOP
PUSH DWORD hbmBitMap ;Bit-map handle from which the pointer image is created
PUSH WORD fPointerSize ;Pointer-size indicator
PUSH WORD sxHotspot ;x offset of hotspot within pointer from its lower left corner (in pels)
PUSH WORD syHotspot ;y offset of hotspot within pointer from its lower left corner (in pels)

CALL WinCreatePointer

Returns DWORD hptr ;Pointer handle
```

WinCreatePointerIndirect

```
EXTRN WinCreatePointerIndirect:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndDesktop ;Desktop-window handle or HWND_DESKTOP
PUSH@ OTHER ptriPointerInfo ;Pointer information structure
CALL WinCreatePointerIndirect

Returns DWORD hptr ;Pointer handle
```

WinCreateStdWindow

```
EXTRN WinCreateStdWindow:FAR
INCL_WINFRAMEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hwndParent  ;Parent-window handle
PUSH  DWORD  fStyle      ;Frame-window style
PUSH@  DWORD  fCreateFlags ;Frame-creation flags
PUSH@  ASCIIZ  szClassClient ;Client-window class name
PUSH@  ASCIIZ  szTitle    ;Title-bar text
PUSH  DWORD  fStyleClient ;Client-window style
PUSH  WORD   Resource     ;Resource identifier
PUSH  WORD   Id           ;Frame-window identifier
PUSH@  DWORD  hwndClient  ;Client-window handle
CALL  WinCreateStdWindow

Returns DWORD  hwndFrame    ;Frame-window handle
```

WinCreateSwitchEntry

```
EXTRN WinCreateSwitchEntry:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH@  OTHER  swctlSwitchData ;Switch data
CALL  WinCreateSwitchEntry

Returns DWORD  hswitchSwitch ;Handle to the newly-created switch list entry
```

WinCreateWindow

```
EXTRN WinCreateWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndParent ;Parent-window handle
PUSH@  ASCIIZ  szClassName ;Registered-class name
PUSH@  ASCIIZ  szName     ;Window text
PUSH  DWORD  fStyle      ;Window style
PUSH  WORD   sxcoord     ;x coordinate of window position
PUSH  WORD   sycoord     ;y coordinate of window position
PUSH  WORD   sWidth      ;Width of window, in window coordinates
PUSH  WORD   sHeight     ;Height of window, in window coordinates
PUSH  DWORD  hwndOwner   ;Owner-window handle
PUSH  DWORD  hwndBehind  ;Sibling-window handle
PUSH  WORD   Id          ;Window identifier
PUSH@  OTHER  CtlData    ;Control data
PUSH@  OTHER  PresParams ;Presentation parameters
CALL  WinCreateWindow

Returns DWORD  hwnd        ;Window handle
```

WinDdeInitiate

```
EXTRN WinDdeInitiate:FAR
INCL_WINDDE equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndClient ;Client's window handle
PUSH@  ASCIIZ  szAppName ;Application name
PUSH@  ASCIIZ  szTopicName ;Topic name
CALL  WinDdeInitiate

Returns WORD  fSuccess    ;Success indicator
```


WinDdePostMsg

```
EXTRN WinDdePostMsg:FAR
INCL_WINDDE equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndTo    ;Window handle of target
PUSH  DWORD  hwndFrom  ;Window handle of originator
PUSH  WORD   usMsgId   ;Message identifier
PUSH@ OTHER  ddeData   ;DDE structure
PUSH  WORD   fRetry    ;Retry indicator
CALL  WinDdePostMsg

Returns WORD  fSuccess ;Success indicator
```

WinDdeRespond

```
EXTRN WinDdeRespond:FAR
INCL_WINDDE equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndClient ;Client's window handle
PUSH  DWORD  hwndServer ;Server's window handle
PUSH@ ASCIIZ  szAppName ;Application name
PUSH@ ASCIIZ  szTopicName ;Topic name
CALL  WinDdeRespond

Returns DWORD  mresReply ;Message return data
```

WinDefAVioWindowProc

```
EXTRN WinDefAVioWindowProc:FAR
INCL_AVIO equ 1 ;Or use INCL_PM

PUSH  DWORD  hwnd      ;Window handle
PUSH  WORD   usMsgid   ;Message identity
PUSH  DWORD  mpParam1  ;Parameter 1
PUSH  DWORD  mpParam2  ;Parameter 2
CALL  WinDefAVioWindowProc

Returns DWORD  mresreply ;Message return data
```

WinDefDlgProc

```
EXTRN WinDefDlgProc:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hwndDlg   ;Dialog-window handle
PUSH  WORD   usMsgid   ;Message identity
PUSH  DWORD  mpParam1  ;Parameter 1
PUSH  DWORD  mpParam2  ;Parameter 2
CALL  WinDefDlgProc

Returns DWORD  mresReply ;Message-return data
```

WinDefWindowProc

```
EXTRN WinDefWindowProc:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hwnd      ;Window handle
PUSH  WORD   usMsgid   ;Message identity
PUSH  DWORD  mpParam1  ;Parameter 1
PUSH  DWORD  mpParam2  ;Parameter 2
CALL  WinDefWindowProc

Returns DWORD  mresReply ;Message-return data
```

WinDeleteAtom

```
EXTRN WinDeleteAtom:FAR
INCL_WINATOM equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hatomtblAtomTbl ;Atom-table handle
PUSH WORD atom ;Atom identifying the atom to be deleted
CALL WinDeleteAtom

Returns WORD ReturnCode ;Return code
```

WinDeleteLibrary

```
EXTRN WinDeleteLibrary:FAR
INCL_WINLOAD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hlibLibhandle ;Library handle to be deleted
CALL WinDeleteLibrary

Returns WORD fDeleted ;Library-deleted indicator
```

WinDeleteProcedure

```
EXTRN WinDeleteProcedure:FAR
INCL_WINLOAD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER wndproc ;Window procedure identifier to be deleted
CALL WinDeleteProcedure

Returns WORD fsuccess ;Procedure-deleted indicator
```

WinDestroyAccelTable

```
EXTRN WinDestroyAccelTable:FAR
INCL_WINACCELERATORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD haccelAccel ;Accelerator-table handle
CALL WinDestroyAccelTable

Returns WORD fSuccess ;Success indicator
```

WinDestroyAtomTable

```
EXTRN WinDestroyAtomTable:FAR
INCL_WINATOM equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hatomtblAtomTbl ;Atom-table handle
CALL WinDestroyAtomTable

Returns DWORD hatomtblReturnCode ;Return code
```

WinDestroyCursor

```
EXTRN WinDestroyCursor:FAR
INCL_WINCURSORS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Window handle to which the cursor belongs
CALL WinDestroyCursor

Returns WORD fSuccess ;Success indicator
```

WinDestroyHeap

```
EXTRN WinDestroyHeap:FAR
INCL_WINHEAP equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hheapHeap ;Heap handle
CALL WinDestroyHeap

Returns DWORD hheapHeapRet ;Returned heap handle
```

WinDestroyMsgQueue

```
EXTRN WinDestroyMsgQueue:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hmq ;Message-queue handle
CALL WinDestroyMsgQueue

Returns WORD fDestroyed ;Queue-destroyed indicator
```

WinDestroyPointer

```
EXTRN WinDestroyPointer:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hpPtrPointer ;Handle of pointer to be destroyed
CALL WinDestroyPointer

Returns WORD fSuccess ;Success indicator
```

WinDestroyWindow

```
EXTRN WinDestroyWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Window handle
CALL WinDestroyWindow

Returns WORD fSuccess ;Window-destroyed indicator
```

WinDismissDlg

```
EXTRN WinDismissDlg:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwndDlg ;Dialog-window handle
PUSH WORD usResult ;Reply value
CALL WinDismissDlg

Returns WORD fSuccess ;Dialog-dismissed indicator
```

WinDispatchMsg

```
EXTRN WinDispatchMsg:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER qmsgMsg ;Message structure
CALL WinDispatchMsg

Returns DWORD mresReply ;Message-return data
```

WinDlgBox

```
EXTRN WinDlgBox:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwndParent ;Parent-window handle of the created dialog window
PUSH DWORD hwndOwner ;Requested owner-window handle of the created dialog window
PUSH@ OTHER DlgProc ;Dialog procedure for the created dialog window
PUSH WORD Resource ;Resource identity containing the dialog template
PUSH WORD usDlgid ;Dialog-template identity within the resource file
PUSH@ OTHER CreateParams ;Application-defined data area
CALL WinDlgBox

Returns WORD usResult ;Reply value
```

WinDrawBitmap

```
EXTRN WinDrawBitmap:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hps      ;Handle of presentation space in which the bit map is drawn
PUSH  DWORD  hbm      ;Bit-map handle
PUSH@  OTHER  rcIsrc   ;Subrectangle of bit map to be drawn
PUSH@  OTHER  ptIDest  ;Bit-map destination
PUSH  DWORD  lForeColor ;Foreground color
PUSH  DWORD  lBackColor ;Background color
PUSH  WORD   fsRgf     ;Flags that determine how the bit map is drawn
CALL  WinDrawBitmap

Returns WORD  fSuccess ;Success indicator
```

WinDrawBorder

```
EXTRN WinDrawBorder:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH@  OTHER  rcIRectangle ;Bounding rectangle for the border
PUSH  WORD   sVertSideWidth ;Width of border rectangle vertical sides
PUSH  WORD   sHorizSideWidth ;Width of border rectangle horizontal sides
PUSH  DWORD  lBorderColor ;Color of edge of border
PUSH  DWORD  lInteriorColor ;Color of interior of border
PUSH  WORD   fsCmd     ;Flags controlling the way in which the border is drawn
CALL  WinDrawBorder

Returns WORD  fSuccess ;Success indicator
```

WinDrawPointer

```
EXTRN WinDrawPointer:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle into which the pointer is drawn
PUSH  WORD   sx       ;X coordinate at which to draw the pointer, in device coordinates
PUSH  WORD   sy       ;Y coordinate at which to draw the pointer, in device coordinates
PUSH  DWORD  hpPtrPointer ;Pointer handle
PUSH  WORD   usHalftone ;Shading control
CALL  WinDrawPointer

Returns WORD  fSuccess ;Success indicator
```

WinDrawText

```
EXTRN WinDrawText:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hps      ;Presentation-space handle
PUSH  WORD   sCount   ;Count of the number of characters in the string
PUSH@  OTHER  chText   ;Character string to be drawn
PUSH@  OTHER  rcIRectangle ;Text rectangle
PUSH  DWORD  lForeColor ;Foreground color
PUSH  DWORD  lBackColor ;Background color
PUSH  WORD   fsCmd     ;An array of flags that determines how the text is drawn
CALL  WinDrawText

Returns WORD  sChars ;Count of characters drawn within the rectangle
```

WinEmptyClipbrd

```
EXTRN WinEmptyClipbrd:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
CALL WinEmptyClipbrd

Returns WORD fSuccess ;Success indicator
```

WinEnablePhysInput

```
EXTRN WinEnablePhysInput:FAR
INCL_WININPUT equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndDesktop ;Desktop-window handle
PUSH WORD fNewInputState ;New state for the queuing of physical input
CALL WinEnablePhysInput

Returns WORD fOldInputState ;Previous state for the queuing of physical input
```

WinEnableWindow

```
EXTRN WinEnableWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH WORD fNewEnabled ;New enabled state
CALL WinEnableWindow

Returns WORD fSuccess ;Window enabled indicator
```

WinEnableWindowUpdate

```
EXTRN WinEnableWindowUpdate:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH WORD fNewVisibility ;New visibility state
CALL WinEnableWindowUpdate

Returns WORD fSuccess ;Visibility-changed indicator
```

WinEndEnumWindows

```
EXTRN WinEndEnumWindows:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD henum ;Enumeration handle
CALL WinEndEnumWindows

Returns WORD fSuccess ;Success indicator
```

WinEndPaint

```
EXTRN WinEndPaint:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
CALL WinEndPaint

Returns WORD fSuccess ;Success indicator
```

WinEnumClipbrdFmts

```
EXTRN WinEnumClipbrdFmts:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH WORD usPrev ;Previous clipboard-data format index
CALL WinEnumClipbrdFmts

Returns WORD usNext ;Next clipboard-data format index
```

WinEnumDlgItem

```
EXTRN WinEnumDlgItem:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnDlG ;Dialog-window handle
PUSH DWORD hwnD ;Child-window handle
PUSH WORD usCode ;Item-type code
PUSH WORD fLock ;Lock indicator
CALL WinEnumDlgItem

Returns DWORD hwnDItem ;Item-window handle
```

WinEqualRect

```
EXTRN WinEqualRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER rc1Rect1 ;First rectangle
PUSH@ OTHER rc1Rect2 ;Second rectangle
CALL WinEqualRect

Returns WORD fEqual ;Equality indicator
```

WinExcludeUpdateRegion

```
EXTRN WinExcludeUpdateRegion:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hps ;Presentation-space handle whose clipping region is to be updated
PUSH DWORD hwnD ;Window handle
CALL WinExcludeUpdateRegion

Returns WORD sComplexity ;Complexity value
```

WinFillRect

```
EXTRN WinFillRect:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rc1Rect ;Rectangle to be filled, in window coordinates
PUSH DWORD lColor ;Color with which to fill the rectangle
CALL WinFillRect

Returns WORD fSuccess ;Success indicator
```

WinFindAtom

```
EXTRN WinFindAtom:FAR
INCL_WINATOM equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hatomtblAtomTbl ;Atom-table handle
PUSH@ ASCII szAtomName ;Atom name
CALL WinFindAtom

Returns WORD atom ;Atom value
```

WinFlashWindow

```
EXTRN WinFlashWindow:FAR
INCL_WINFRAMEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnD ;Handle of window to be flashed
PUSH WORD fFlash ;Start-flashing indicator
CALL WinFlashWindow

Returns WORD fSuccess ;Success indicator
```

WinFocusChange

```
EXTRN WinFocusChange:FAR
INCL_WININPUT equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnDskTop ;Desktop-window handle
PUSH DWORD hwnDNewFocus ;Window handle to receive the focus
PUSH WORD fsFocusChange ;Focus changing indicators
CALL WinFocusChange

Returns WORD fSuccess ;Success indicator
```

WinFreeErrorInfo

```
EXTRN WinFreeErrorInfo:FAR
INCL_WINERRORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH@ OTHER erriErrorInfo ;Error-information block whose memory is to be released
CALL WinFreeErrorInfo

Returns WORD fSuccess ;Success indicator
```

WinFreeMem

```
EXTRN WinFreeMem:FAR
INCL_WINHEAP equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hheapHeap ;Handle to a heap
PUSH WORD npbMem ;Memory block to be freed
PUSH WORD usLength ;Size of the memory to be freed
CALL WinFreeMem

Returns WORD npbReturn ;Values as follows
```

WinGetClipPS

```
EXTRN WinGetClipPS:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnD ;Handle of window for which the presentation space is required
PUSH DWORD hwnDClipWindow ;Handle of window for clipping
PUSH WORD usClipflags ;Clipping control flags
CALL WinGetClipPS

Returns DWORD hps ;Presentation-space handle that can be used for drawing
```

WinGetCurrentTime

```
EXTRN WinGetCurrentTime:FAR
INCL_WINTIMER equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
CALL WinGetCurrentTime

Returns DWORD ulTime ;System-timer count
```

WinGetDlgMsg

```
EXTRN WinGetDlgMsg:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnDDlg ;Dialog-window handle
PUSH@ OTHER qmsgmsg ;Message structure
CALL WinGetDlgMsg

Returns WORD fResult ;Continue message indicator
```

WinGetErrorInfo

```
EXTRN WinGetErrorInfo:FAR
INCL_WINERRORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
CALL WinGetErrorInfo

Returns @OTHER erriErrorInfo ;Error information
```

WinGetKeyState

```
EXTRN WinGetKeyState:FAR
INCL_WININPUT equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndDesktop ;Desktop-window handle:
PUSH WORD sVk ;Virtual key value
CALL WinGetKeyState

Returns WORD sKeyState ;Key state
```

WinGetLastError

```
EXTRN WinGetLastError:FAR
INCL_WINERRORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
CALL WinGetLastError

Returns DWORD erridErrorCode ;Last-error state
```

WinGetMinPosition

```
EXTRN WinGetMinPosition:FAR
INCL_WINFRAMEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Frame-window handle
PUSH@ OTHER Swp ;Set window position structure
PUSH@ OTHER ptlPoint ;Preferred position
CALL WinGetMinPosition

Returns WORD fSuccess ;Success indicator
```

WinGetMsg

```
EXTRN WinGetMsg:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER qmsgmsg ;Message structure
PUSH DWORD hwndFilter ;Window filter
PUSH WORD usFirst ;First message identity
PUSH WORD usLast ;Last message identity
CALL WinGetMsg

Returns WORD fResult ;Continue message indicator
```

WinGetNextWindow

```
EXTRN WinGetNextWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD henum ;Enumeration handle
CALL WinGetNextWindow

Returns DWORD hwndNext ;Next window handle in enumeration list
```


WinGetPhysKeyState

```
EXTRN WinGetPhysKeyState:FAR
INCL_WININPUT equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnDskTop ;Desktop-window handle
PUSH WORD sScancode ;Hardware scancode
CALL WinGetPhysKeyState

Returns WORD sKeyState ;Key state
```

WinGetPS

```
EXTRN WinGetPS:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnD ;Handle of window for which the presentation space is required
CALL WinGetPS

Returns DWORD hps ;Presentation-space handle that can be used for drawing in the window
```

WinGetScreenPS

```
EXTRN WinGetScreenPS:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnDskTop ;Desktop-window handle
CALL WinGetScreenPS

Returns DWORD hpsScreenPS ;Presentation-space handle
```

WinGetSysBitmap

```
EXTRN WinGetSysBitmap:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnDskTop ;Desktop-window handle
PUSH WORD iIndex ;System bit-map index value
CALL WinGetSysBitmap

Returns DWORD hbm ;System bit-map handle
```

WinInflateRect

```
EXTRN WinInflateRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER rclrect ;Rectangle to be expanded
PUSH WORD scx ;Horizontal expansion
PUSH WORD scy ;Vertical expansion
CALL WinInflateRect

Returns WORD fSuccess ;Success indicator
```

WinInitialize

```
EXTRN WinInitialize:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH WORD fsOptions ;Initialization options
CALL WinInitialize

Returns DWORD hab ;Anchor-block handle
```

WinInSendMessage

```
EXTRN WinInSendMessage:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
CALL WinInSendMessage

Returns WORD fProcessing ;Message-processing indicator
```

WinInstStartApp

```
EXTRN WinInstStartApp:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hini ;Initialization-file handle
PUSH DWORD hwndNotifyWindow ;Notification-window handle
PUSH WORD cCount ;Count of elements in the Application parameter. Must be 1 or
2
PUSH@ ASCIIZ aszApplication ;Identifier of the application to be started
PUSH@ ASCIIZ szCmdLine ;Input parameters for the application to be started
PUSH@ OTHER Data ;Start data
PUSH WORD fsOptions ;Option indicators
CALL WinInstStartApp

Returns DWORD happ ;Application handle
```

WinIntersectRect

```
EXTRN WinIntersectRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER rcIDest ;Intersection rectangle
PUSH@ OTHER rcIRect1 ;First rectangle
PUSH@ OTHER rcIRect2 ;Second rectangle
CALL WinIntersectRect

Returns WORD fSuccess ;Success indicator
```

WinInvalidateRect

```
EXTRN WinInvalidateRect:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Handle of window whose update region is to be changed
PUSH@ OTHER rcIPrc ;Update rectangle
PUSH WORD fIncludeClippedChildren ;Invalidation-scope indicator
CALL WinInvalidateRect

Returns WORD fSuccess ;Success indicator
```

WinInvalidateRegion

```
EXTRN WinInvalidateRegion:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Handle of window whose update region is to be changed
PUSH DWORD hrgn ;Handle of the region to be added to the window's
update region
PUSH WORD fIncludeClippedChildren ;Invalidation-scope indicator
CALL WinInvalidateRegion

Returns WORD fSuccess ;Success indicator
```

WinInvertRect

```
EXTRN WinInvertRect:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER rcRect ;Rectangle to be inverted
CALL WinInvertRect

Returns WORD fSuccess ;Success indicator
```

WinIsChild

```
EXTRN WinIsChild:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndChild ;Child-window handle
PUSH DWORD hwndParent ;Parent-window handle
CALL WinIsChild

Returns WORD fRelated ;Related indicator
```

WinIsRectEmpty

```
EXTRN WinIsRectEmpty:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER rcIprc ;Rectangle
CALL WinIsRectEmpty

Returns WORD fEmpty ;Empty indicator
```

WinIsThreadActive

```
EXTRN WinIsThreadActive:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle of calling thread
CALL WinIsThreadActive

Returns WORD fActive ;Active-window indicator
```

WinIsWindow

```
EXTRN WinIsWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hwnd ;Window handle
CALL WinIsWindow

Returns WORD fValid ;Validity indicator
```

WinIsWindowEnabled

```
EXTRN WinIsWindowEnabled:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
CALL WinIsWindowEnabled

Returns WORD fEnabled ;Enabled-state indicator
```

WinIsWindowShowing

```
EXTRN WinIsWindowShowing:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Window handle
CALL WinIsWindowShowing

Returns WORD fShowing ;Showing state indicator
```

WinIsWindowVisible

```
EXTRN WinIsWindowVisible:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
CALL WinIsWindowVisible

Returns WORD fVisible ;Visibility-state indicator
```

WinLoadAccelTable

```
EXTRN WinLoadAccelTable:FAR
INCL_WINACCELERATORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH WORD Resource ;Resource identity containing the accelerator table
PUSH WORD idAccelTable ;Accelerator-table identifier, within the resource file
CALL WinLoadAccelTable

Returns DWORD haccclAccel ;Accelerator-table handle
```

WinLoadDlg

```
EXTRN WinLoadDlg:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwndParent ;Parent-window handle of the created dialog window
PUSH DWORD hwndOwner ;Requested owner-window handle of the created dialog window
PUSH@ OTHER DlgProc ;Dialog procedure for the created dialog window
PUSH WORD Resource ;Resource identity containing the dialog template
PUSH WORD idDlgid ;Dialog-template identity within the resource file
PUSH@ OTHER CreateParams ;Application-defined data area
CALL WinLoadDlg

Returns DWORD hwndDlg ;Dialog-window handle
```

WinLoadLibrary

```
EXTRN WinLoadLibrary:FAR
INCL_WINLOAD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCIIZ szLibname ;Library name
CALL WinLoadLibrary

Returns DWORD hlibLibhandle ;Library handle
```

WinLoadMenu

```
EXTRN WinLoadMenu:FAR
INCL_WINMENUS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndOwner ;Owner- and parent-window handle
PUSH WORD Resource ;Resource identifier
PUSH WORD idMenuid ;Menu identifier within the resource file
CALL WinLoadMenu

Returns DWORD hwndMenu ;Menu-window handle
```

WinLoadPointer

```
EXTRN WinLoadPointer:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndDeskTop ;Desktop-window handle
PUSH WORD Resource ;Resource identity containing the pointer definition
PUSH WORD idPointer ;Identifier of the pointer to be loaded
CALL WinLoadPointer

Returns DWORD hptr ;Pointer handle
```

WinLoadProcedure

```
EXTRN WinLoadProcedure:FAR
INCL_WINLOAD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  DWORD  hlibLibhandle ;Library handle
PUSH@  ASCIIZ  szProcname ;Procedure name
CALL  WinLoadProcedure

Returns @OTHER Wndproc ;Window-procedure identifier
```

WinLoadString

```
EXTRN WinLoadString:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   Resource  ;Resource identity containing the string
PUSH  WORD   idString  ;String identifier
PUSH  WORD   sBufferMax ;Size of buffer
PUSH@  ASCIIZ  szBuffer ;Buffer that is to receive the string
CALL  WinLoadString

Returns WORD  sLength ;The length of the string returned
```

WinLockHeap

```
EXTRN WinLockHeap:FAR
INCL_WINHEAP equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hheapHeap ;Handle to a heap
CALL  WinLockHeap

Returns @OTHER Start ;Segment containing the passed heap
```

WinLockVisRegions

```
EXTRN WinLockVisRegions:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDesktop ;Desktop-window handle or HWND_DESKTOP
PUSH  WORD   fLock       ;Indicates whether the visible regions are being locked or
                        unlocked
CALL  WinLockVisRegions

Returns WORD  fSuccess ;Success indicator
```

WinLockWindow

```
EXTRN WinLockWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd ;Window handle
PUSH  WORD   fLock ;Lock indicator
CALL  WinLockWindow

Returns DWORD hwndRet ;Locked-window handle
```

WinLockWindowUpdate

```
EXTRN WinLockWindowUpdate:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDesktop ;Desktop handle of the screen containing the window to be locked
PUSH  DWORD  hwndLockUpdate ;Handle of window in which output is to be prevented
CALL  WinLockWindowUpdate

Returns WORD  fSuccess ;Success indicator
```

WinMakePoints

```
EXTRN WinMakePoints:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  wptppt    ;Points to be converted
PUSH  WORD   ccount    ;Number of points to be converted
CALL  WinMakePoints

Returns WORD  fSuccess ;Success indicator
```

WinMakeRect

```
EXTRN WinMakeRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  wrcprc    ;Rectangle to be converted
CALL  WinMakeRect

Returns WORD  fSuccess ;Success indicator
```

WinMapDlgPoints

```
EXTRN WinMapDlgPoints:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDlg   ;Dialog-window handle
PUSH@ OTHER  aptlPoints ;Coordinate points to be mapped
PUSH  WORD   usCount    ;Number of coordinate points
PUSH  WORD   fOptions   ;Calculation control
CALL  WinMapDlgPoints

Returns WORD  fSuccess ;Coordinates-mapped indicator
```

WinMapWindowPoints

```
EXTRN WinMapWindowPoints:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndFrom  ;Handle of the window from whose coordinates points are to be
                        mapped
PUSH  DWORD  hwndTo    ;Handle of the window to whose coordinates points are to be mapped
PUSH@ OTHER  aptlPoints ;Points to be mapped to the new coordinate system
PUSH  WORD   sCount    ;Number of points to be mapped
CALL  WinMapWindowPoints

Returns WORD  fSuccess ;Success indicator
```

WinMessageBox

```
EXTRN WinMessageBox:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hwndParent ;Parent-window handle of the created message-box window
PUSH  DWORD  hwndOwner  ;Requested owner-window handle of the created message-box window
PUSH@ ASCIIZ  szText     ;Message-box window message
PUSH@ ASCIIZ  szTitle    ;Message-box window title
PUSH  WORD   usWindow    ;Message-box window identity
PUSH  WORD   fsStyle     ;Message-box window style
CALL  WinMessageBox

Returns WORD  usResponse ;User-response value
```

WinMsgMuxSemWait

```
EXTRN WinMsgMuxSemWait:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH@ WORD usIndexNbr ;Index of the cleared semaphore in the list
PUSH@ OTHER List ;List of semaphore descriptors
PUSH DWORD dtTimeout ;Function time-out value
CALL WinMsgMuxSemWait

Returns WORD usrc ;Return code
```

WinMsgSemWait

```
EXTRN WinMsgSemWait:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hsem ;Semaphore handle
PUSH DWORD dtTimeout ;Function time-out value
CALL WinMsgSemWait

Returns WORD usrc ;Return code
```

WinMultWindowFromIDs

```
EXTRN WinMultWindowFromIDs:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndParent ;Parent-window handle
PUSH@ OTHER ahwnd ;Window handles
PUSH WORD usFirst ;First window identity value in the range (inclusive)
PUSH WORD usLast ;Last window identity value in the range (inclusive)
CALL WinMultWindowFromIDs

Returns WORD sWindows ;Number of window handles returned
```

WinNextChar

```
EXTRN WinNextChar:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH WORD usCodepage ;Code page
PUSH WORD usCountry ;Country code
PUSH@ ASCIIZ szCurrentChar ;Current character in a null-terminated string
CALL WinNextChar

Returns @ASCIIZ szNextChar ;Next character in the null-terminated string
```

WinOffsetRect

```
EXTRN WinOffsetRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER rcIrect ;Rectangle to be offset
PUSH WORD scx ;x value of offset
PUSH WORD scy ;y value of offset
CALL WinOffsetRect

Returns WORD fSuccess ;Success indicator
```

WinOpenClipbrd

```
EXTRN WinOpenClipbrd:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
CALL WinOpenClipbrd

Returns WORD fSuccess ;Success indicator
```

WinOpenWindowDC

```
EXTRN WinOpenWindowDC:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Window handle
CALL WinOpenWindowDC

Returns DWORD hdc ;Device-context handle
```

WinPeekMsg

```
EXTRN WinPeekMsg:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER qmsgmsg ;Message structure
PUSH DWORD hwndFilter ;Window filter
PUSH WORD usFirst ;First message identity
PUSH WORD usLast ;Last message identity
PUSH WORD fsOptions ;Options
CALL WinPeekMsg

Returns WORD fResult ;Message-available indicator
```

WinPostMsg

```
EXTRN WinPostMsg:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Window handle
PUSH WORD usMsgid ;Message identity
PUSH DWORD mpParam1 ;Parameter 1
PUSH DWORD mpParam2 ;Parameter 2
CALL WinPostMsg

Returns WORD fResult ;Message-posted indicator
```

WinPostQueueMsg

```
EXTRN WinPostQueueMsg:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hmq ;Message-queue handle
PUSH WORD usMsgId ;Message identifier
PUSH DWORD mpParam1 ;Parameter 1
PUSH DWORD mpParam2 ;Parameter 2
CALL WinPostQueueMsg

Returns WORD fSuccess ;Success indicator
```

WinPrevChar

```
EXTRN WinPrevChar:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH WORD usCodepage ;Code page
PUSH WORD usCountry ;Country code
PUSH@ ASCIIZ szStart ;Character string that contains CurrentChar
PUSH@ ASCIIZ szCurrentChar ;Current character
CALL WinPrevChar

Returns @ASCIIZ szPrevChar ;Previous character
```


WinProcessDlg

```
EXTRN WinProcessDlg:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnDlG ;Dialog-window handle
CALL WinProcessDlg

Returns WORD usResult ;Reply value
```

WinPtInRect

```
EXTRN WinPtInRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER rcIrect ;Rectangle to be queried
PUSH@ OTHER ptIpoint ;Point to be queried
CALL WinPtInRect

Returns WORD f ;Success indicator
```

WinQueryAccelTable

```
EXTRN WinQueryAccelTable:FAR
INCL_WINACCELERATORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hwnDFrame ;Frame-window handle
CALL WinQueryAccelTable

Returns DWORD haccelAccel ;Accelerator-table handle
```

WinQueryActiveWindow

```
EXTRN WinQueryActiveWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnDParent ;Parent-window handle for which the active window is required
PUSH WORD fLock ;Windows lock-state indicator
CALL WinQueryActiveWindow

Returns DWORD hwnDActive ;Active-window handle
```

WinQueryAnchorBlock

```
EXTRN WinQueryAnchorBlock:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnD ;Window handle
CALL WinQueryAnchorBlock

Returns DWORD hab ;Anchor block handle
```

WinQueryAtomLength

```
EXTRN WinQueryAtomLength:FAR
INCL_WINATOM equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hatomTblAtomTbl ;Atom-table handle
PUSH WORD atom ;Atom whose associated character-string length is to be
returned

CALL WinQueryAtomLength

Returns WORD usretlen ;String length
```

WinQueryAtomName

```
EXTRN WinQueryAtomName:FAR
INCL_WINATOM equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hatomtblAtomTbl ;Atom-table handle
PUSH  WORD   atom            ;Atom; identifies the character string to be retrieved
PUSH@  ASCIIZ  szBuffer      ;Buffer to receive the character string
PUSH  WORD   usBufferMax     ;Buffer size in bytes
CALL  WinQueryAtomName

Returns WORD  usretlen      ;Length of retrieved character string
```

WinQueryAtomUsage

```
EXTRN WinQueryAtomUsage:FAR
INCL_WINATOM equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hatomtblAtomTbl ;Atom-table handle
PUSH  WORD   atom            ;Atom whose use count is to be returned
CALL  WinQueryAtomUsage

Returns WORD  uscount      ;Use count of the atom
```

WinQueryCapture

```
EXTRN WinQueryCapture:FAR
INCL_WININPUT equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDesktop ;Desktop-window handle
PUSH  WORD   fLock       ;Lock-request indicator
CALL  WinQueryCapture

Returns DWORD  hwnd      ;Handle of the window with the pointer captured
```

WinQueryClassInfo

```
EXTRN WinQueryClassInfo:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH@  ASCIIZ  szClassName ;Class name
PUSH@  OTHER  clsClassInfo ;Class information structure
CALL  WinQueryClassInfo

Returns WORD  fExists      ;Class-exists indicator
```

WinQueryClassName

```
EXTRN WinQueryClassName:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd ;Window handle
PUSH  WORD   sLength ;Length of Buffer
PUSH@  OTHER  chBuffer ;Class name
CALL  WinQueryClassName

Returns WORD  sRetLen ;Returned class name length
```

WinQueryClipbrdData

```
EXTRN WinQueryClipbrdData:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab ;Anchor-block handle
PUSH  WORD   usfmt ;Format of the data to be accessed
CALL  WinQueryClipbrdData

Returns DWORD  ulData ;Handle to the clipboard data
```

WinQueryClipbrdFmtInfo

```
EXTRN WinQueryClipbrdFmtInfo:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   usfmt     ;Format of the data to be queried
PUSH@ WORD   usFmtInfo ;Memory model and usage flags
CALL  WinQueryClipbrdFmtInfo

Returns WORD  fExists  ;Format-exists indicator
```

WinQueryClipbrdOwner

```
EXTRN WinQueryClipbrdOwner:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   fLock     ;Lock state indicator of clipboard owner window
CALL  WinQueryClipbrdOwner

Returns DWORD hwndClipbrdOwner ;Window handle of the current clipboard owner
```

WinQueryClipbrdViewer

```
EXTRN WinQueryClipbrdViewer:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   fLock     ;Lock-state indicator of clipboard viewer window
CALL  WinQueryClipbrdViewer

Returns DWORD hwndClipbrdViewer ;Current clipboard viewer window handle
```

WinQueryCp

```
EXTRN WinQueryCp:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hmq      ;Message queue
CALL  WinQueryCp

Returns WORD  usCodePage ;Code page
```

WinQueryCpList

```
EXTRN WinQueryCpList:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   uscount   ;Maximum number of code pages returned
PUSH@ OTHER  ausCodepage ;Code page list
CALL  WinQueryCpList

Returns WORD  usTotCount ;Total number of code pages available
```

WinQueryCursorInfo

```
EXTRN WinQueryCursorInfo:FAR
INCL_WINCURSORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDeskTop ;Desktop-window handle
PUSH@ OTHER  csriCursorInfo ;Cursor information
CALL  WinQueryCursorInfo

Returns WORD  fCursor    ;Current-cursor indicator
```

WinQueryDefinition

```
EXTRN WinQueryDefinition:FAR
INCL_WINPROGRAMLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH  DWORD  hprogProgHandle ;Handle of the program whose information is to be returned
PUSH@ OTHER  pibProgramInfo ;Program-information data
PUSH  WORD   usMaxLength   ;Maximum length in bytes
CALL  WinQueryDefinition

Returns WORD   usLength      ;Length of returned data
```

WinQueryDesktopWindow

```
EXTRN WinQueryDesktopWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH  DWORD  hdc          ;Device-context handle
CALL  WinQueryDesktopWindow

Returns DWORD  hwndDeskTop ;Desktop-window handle
```

WinQueryDlgItemShort

```
EXTRN WinQueryDlgItemShort:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hwndDlg ;Parent-window handle
PUSH  WORD   idItem  ;Identity of the child window whose text is to be converted
PUSH@ WORD   sResult ;Integer value resulting from the conversion
PUSH  WORD   fSigned ;Sign indicator
CALL  WinQueryDlgItemShort

Returns WORD   fSuccess ;Success indicator
```

WinQueryDlgItemText

```
EXTRN WinQueryDlgItemText:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hwndDlg ;Parent-window handle
PUSH  WORD   idItem  ;Identity of the child window whose text is to be queried
PUSH  WORD   sMaxText ;Length of Text
PUSH@ ASCIIZ  szText ;Output string
CALL  WinQueryDlgItemText

Returns WORD   usRetLen ;Actual number of characters returned
```

WinQueryDlgItemTextLength

```
EXTRN WinQueryDlgItemTextLength:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hwndDlg ;Parent-window handle
PUSH  WORD   idItem  ;Identity of the child window whose text is to be queried
CALL  WinQueryDlgItemTextLength

Returns WORD   sRetLen ;Length of text
```

WinQueryFocus

```
EXTRN WinQueryFocus:FAR
INCL_WININPUT equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDeskTop ;Desktop-window handle
PUSH  WORD   fLock        ;Window lock state indicator
CALL  WinQueryFocus

Returns DWORD  hwndFocus   ;Focus-handle
```

WinQueryMsgPos

```
EXTRN WinQueryMsgPos:FAR
INCL_WINMESSAGEGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  ptlptrpos ;Pointer position in screen coordinates
CALL  WinQueryMsgPos

Returns WORD  fSuccess ;Success indicator
```

WinQueryMsgTime

```
EXTRN WinQueryMsgTime:FAR
INCL_WINMESSAGEGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
CALL  WinQueryMsgTime

Returns DWORD ulTime ;Time in milliseconds
```

WinQueryObjectWindow

```
EXTRN WinQueryObjectWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDeskTop ;Desktop-window handle
CALL  WinQueryObjectWindow

Returns DWORD hwndObject ;Object-window handle
```

WinQueryPointer

```
EXTRN WinQueryPointer:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDeskTop ;Desktop-window handle
CALL  WinQueryPointer

Returns DWORD hptrPointer ;Pointer handle
```

WinQueryPointerInfo

```
EXTRN WinQueryPointerInfo:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hptr      ;Pointer handle
PUSH@ OTHER  ptriPointerInfo ;Pointer-information structure
CALL  WinQueryPointerInfo

Returns WORD  fSuccess      ;Success indicator
```

WinQueryPointerPos

```
EXTRN WinQueryPointerPos:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDeskTop ;Desktop-window handle
PUSH@ OTHER  ptlPoint     ;Pointer position in screen coordinates
CALL  WinQueryPointerPos

Returns WORD  fSuccess      ;Pointer position returned indicator
```

WinQueryPresParam

```
EXTRN WinQueryPresParam:FAR
INCL_WINSYS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd          ;Window handle
PUSH  DWORD  idAttrType1   ;First attribute type identity
PUSH  DWORD  idAttrType2   ;Second attribute type identity
PUSH@  DWORD  idAttrTypeFound ;Attribute type identity found
PUSH  DWORD  cbAttrValueLen ;Byte count of the size of the AttrValue parameter
PUSH@  OTHER  AttrValue     ;Attribute value
PUSH  WORD   fsOptions      ;Options
CALL  WinQueryPresParam

Returns DWORD  cbRetLen      ;Length of presentation parameter value passed back
```

WinQueryProfileData

```
EXTRN WinQueryProfileData:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH@  ASCIIZ  szAppName   ;Application name
PUSH@  ASCIIZ  szKeyName   ;Key name
PUSH@  OTHER  Value       ;Value data
PUSH@  WORD   usSize      ;Size of value data
CALL  WinQueryProfileData

Returns WORD  fSuccess     ;Success indicator
```

WinQueryProfileInt

```
EXTRN WinQueryProfileInt:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH@  ASCIIZ  szAppName   ;Application name
PUSH@  ASCIIZ  szKeyName   ;Key name
PUSH  WORD   sDefault     ;Default value
CALL  WinQueryProfileInt

Returns WORD  sRequiredInteger ;Key value specified in the intialization file
```

WinQueryProfileSize

```
EXTRN WinQueryProfileSize:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH@  ASCIIZ  szAppName   ;Application name
PUSH@  ASCIIZ  szKeyName   ;Key name
PUSH@  WORD   usValue     ;Data length
CALL  WinQueryProfileSize

Returns WORD  usRetCode    ;Success indicator
```

WinQueryProfileString

```
EXTRN WinQueryProfileString:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH@  ASCIIZ  szAppName   ;Application name
PUSH@  ASCIIZ  szKeyName   ;Key name
PUSH@  ASCIIZ  szDefault   ;Default string
PUSH@  OTHER   ProfileString ;Profile string
PUSH  WORD     usMaxPstring ;Maximum number
CALL  WinQueryProfileString

Returns WORD  usPstring    ;Actual number
```

WinQueryProgramTitles

```
EXTRN WinQueryProgramTitles:FAR
INCL_WINPROGRLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH  DWORD  hprogGroup   ;Handle of the program group whose information is to be returned
PUSH@  OTHER  aprobeBuffer ;Program information buffer
PUSH  WORD    usBufferLen ;Buffer length
PUSH@  WORD    usTotal     ;Total number of structures
CALL  WinQueryProgramTitles

Returns WORD  fSuccess     ;Success indicator
```

WinQueryQueueInfo

```
EXTRN WinQueryQueueInfo:FAR
INCL_WINMESSAGEGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hmq          ;Queue handle
PUSH@  OTHER  mqInfo       ;Message queue information structure to contain the queue information
PUSH  WORD    cbCopied     ;Size of message queue information structure that is provided (in bytes)

CALL  WinQueryQueueInfo

Returns WORD  fSuccess     ;Success indicator
```

WinQueryQueueStatus

```
EXTRN WinQueryQueueStatus:FAR
INCL_WINMESSAGEGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDesktop ;Desktop-window handle
CALL  WinQueryQueueStatus

Returns DWORD fIStatus    ;Status information
```

WinQuerySwitchEntry

```
EXTRN WinQuerySwitchEntry:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hswitchSwitch ;Handle to the task list entry
PUSH@  OTHER  swctlSwitchData ;Switch control data
CALL  WinQuerySwitchEntry

Returns WORD  usRetCode    ;Return code
```

WinQuerySwitchHandle

```
EXTRN WinQuerySwitchHandle:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnid ;Window handle of an application
PUSH WORD idProcess ;Process identity of the application
CALL WinQuerySwitchHandle

Returns DWORD hswitchSwitch ;Switch list handle for the specified application
```

WinQuerySwitchList

```
EXTRN WinQuerySwitchList:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER swblkSwitchEntries ;Switch entries block
PUSH WORD usDataLength ;Maximum length of data returnable in bytes
CALL WinQuerySwitchList

Returns WORD usCount ;Total number of switch list entries present in the system
```

WinQuerySysColor

```
EXTRN WinQuerySysColor:FAR
INCL_WINSYS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnidDesktop ;Desktop-window handle
PUSH DWORD lColor ;System color-index value
PUSH DWORD lReserved ;Reserved
CALL WinQuerySysColor

Returns DWORD lRgbColor ;RGB value
```

WinQuerySysModalWindow

```
EXTRN WinQuerySysModalWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnidDesktop ;Desktop-window handle
PUSH WORD fLock ;Window's lock-state indicator
CALL WinQuerySysModalWindow

Returns DWORD hwnidSysModal ;Handle of system modal window
```

WinQuerySysPointer

```
EXTRN WinQuerySysPointer:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnidDesktop ;Desktop-window handle
PUSH WORD sIdentifier ;System-pointer identifier
PUSH WORD fCopy ;Copy indicator
CALL WinQuerySysPointer

Returns DWORD hptrPointer ;Pointer handle
```

WinQuerySystemAtomTable

```
EXTRN WinQuerySystemAtomTable:FAR
INCL_WINATOM equ 1 ;Or use INCL_WIN or INCL_PM

CALL WinQuerySystemAtomTable

Returns DWORD hatomtblAtomTbl ;System atom-table handle
```


WinQuerySysValue

```
EXTRN WinQuerySysValue:FAR
INCL_WINSYS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnDskTop ;Desktop-window handle
PUSH WORD sValueid ;System-value identity
CALL WinQuerySysValue

Returns DWORD lValue ;System value
```

WinQueryTaskSizePos

```
EXTRN WinQueryTaskSizePos:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH WORD usScreenGroup ;Screen group
PUSH@ OTHER swpPositionData ;Window position and size data
CALL WinQueryTaskSizePos

Returns WORD usRetCode ;Return code
```

WinQueryTaskTitle

```
EXTRN WinQueryTaskTitle:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH WORD usSession ;Session identity of application whose title is requested
PUSH@ ASCIIZ szTitle ;Task title
PUSH WORD usTitlelen ;Maximum length of data returnable in bytes
CALL WinQueryTaskTitle

Returns WORD usRetCode ;Return code
```

WinQueryUpdateRect

```
EXTRN WinQueryUpdateRect:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnD ;Handle of window whose update rectangle is to be queried
PUSH@ OTHER rcIPrC ;Update region that bounds the rectangle (in window coordinates)
CALL WinQueryUpdateRect

Returns WORD fSuccess ;Success indicator
```

WinQueryUpdateRegion

```
EXTRN WinQueryUpdateRegion:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnD ;Handle of window whose update region is to be queried
PUSH DWORD hrGn ;Handle of the window's update region
CALL WinQueryUpdateRegion

Returns WORD sComplexity ;Complexity of resulting region/error indicator
```

WinQueryVersion

```
EXTRN WinQueryVersion:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
CALL WinQueryVersion

Returns DWORD flSysInf ;System information within which the application is operating
```

WinQueryWindow

```
EXTRN WinQueryWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd      ;Handle of window to query
PUSH  WORD   sCode     ;Type of window information
PUSH  WORD   fLock     ;Lock control
CALL  WinQueryWindow

Returns DWORD hwndRelated ;Window handle
```

WinQueryWindowDC

```
EXTRN WinQueryWindowDC:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd ;Window handle
CALL  WinQueryWindowDC

Returns DWORD hdc ;Device-context handle
```

WinQueryWindowLockCount

```
EXTRN WinQueryWindowLockCount:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd ;Window handle
CALL  WinQueryWindowLockCount

Returns WORD  sLock ;Window-lock count
```

WinQueryWindowPos

```
EXTRN WinQueryWindowPos:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd      ;Window handle
PUSH@  OTHER  swp       ;SWP structure
CALL  WinQueryWindowPos

Returns WORD  fSuccess ;Success indicator
```

WinQueryWindowProcess

```
EXTRN WinQueryWindowProcess:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd      ;Window handle
PUSH@  WORD   idpid     ;Process identity of the thread that created the window
PUSH@  WORD   idtid     ;Thread identity of the thread that created the window
CALL  WinQueryWindowProcess

Returns WORD  fSuccess ;Success indicator
```

WinQueryWindowPtr

```
EXTRN WinQueryWindowPtr:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd ;Window handle
PUSH  WORD   sb   ;Index
CALL  WinQueryWindowPtr

Returns @OTHER p ;Pointer value
```

WinQueryWindowRect

```
EXTRN WinQueryWindowRect:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hwnd    ;Window handle
PUSH@ OTHER  rcRect  ;Window rectangle
CALL  WinQueryWindowRect

Returns WORD  fSuccess ;Rectangle-returned indicator
```

WinQueryWindowText

```
EXTRN WinQueryWindowText:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd    ;Window handle
PUSH  WORD   sLength  ;Length
PUSH@ OTHER  chBuffer ;Window text
CALL  WinQueryWindowText

Returns WORD  sRetLen ;Length of returned text
```

WinQueryWindowTextLength

```
EXTRN WinQueryWindowTextLength:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd    ;Window handle
CALL  WinQueryWindowTextLength

Returns WORD  sRetLen ;Length of the window text
```

WinQueryWindowULong

```
EXTRN WinQueryWindowULong:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd    ;Handle of window to be queried
PUSH  WORD   sb      ;Index
CALL  WinQueryWindowULong

Returns DWORD ulValue ;Value contained in the window word
```

WinQueryWindowUShort

```
EXTRN WinQueryWindowUShort:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd    ;Handle of window to be queried
PUSH  WORD   sb      ;Index
CALL  WinQueryWindowUShort

Returns WORD  usValue ;Value contained in the indicated window word
```

WinReallocMem

```
EXTRN WinReallocMem:FAR
INCL_WINHEAP equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hHeap    ;Handle to a heap
PUSH  WORD   npbMemObj ;Memory object
PUSH  WORD   usOld     ;Old size
PUSH  WORD   usNew     ;New size
CALL  WinReallocMem

Returns WORD  npbMem    ;Pointer to memory block
```

WinRegisterClass

```
EXTRN WinRegisterClass:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCIIZ szClassName ;Window-class name
PUSH@ OTHER WndProc ;Window-procedure identifier
PUSH DWORD fClassNameStyle ;Default-window style
PUSH WORD usExtra ;Reserved storage
CALL WinRegisterClass

Returns WORD fRegistered ;Window-class-registration indicator
```

WinRegisterUserDatatype

```
EXTRN WinRegisterUserDatatype:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
PUSH WORD sDatatype ;Data type code to be defined
PUSH WORD sCount ;Number of elements
PUSH@ OTHER asTypes ;Data type codes of structure components
CALL WinRegisterUserDatatype

Returns WORD fSuccess ;Success indicator
```

WinRegisterUserMsg

```
EXTRN WinRegisterUserMsg:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
PUSH WORD usMsgid ;Message identifier
PUSH WORD sType1 ;Datatype of message-parameter 1
PUSH WORD sDir1 ;Direction of message-parameter 1
PUSH WORD sType2 ;Datatype of message-parameter 2
PUSH WORD sDir2 ;Direction of message-parameter 2
PUSH WORD sTypeR ;Datatype of message reply
CALL WinRegisterUserMsg

Returns WORD fSuccess ;Success indicator
```

WinRegisterWindowDestroy

```
EXTRN WinRegisterWindowDestroy:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH WORD fRegister ;Registration indicator
CALL WinRegisterWindowDestroy

Returns WORD fSuccess ;Success indicator
```

WinReleaseHook

```
EXTRN WinReleaseHook:FAR
INCL_WINHOOKS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hmq ;Handle of message queue from which the hook is to be released
PUSH WORD sHook ;Type of hook chain
PUSH@ OTHER Address ;Address of the hook routine
PUSH WORD Module ;Module handle
CALL WinReleaseHook

Returns WORD fSuccess ;Success indicator
```

WinReleasePS

```
EXTRN WinReleasePS:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hps ;Handle of the cache presentation space to release
CALL WinReleasePS

Returns WORD fSuccess ;Success indicator
```

WinRemovePresParam

```
EXTRN WinRemovePresParam:FAR
INCL_WINSYS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH DWORD idAttrType ;Attribute type identity
CALL WinRemovePresParam

Returns WORD fSuccess ;Success indicator
```

WinRemoveSwitchEntry

```
EXTRN WinRemoveSwitchEntry:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hswitchSwitch ;Switch-list (task-list) entry handle
CALL WinRemoveSwitchEntry

Returns WORD usRetCode ;Success indicator
```

WinScrollWindow

```
EXTRN WinScrollWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Window handle
PUSH WORD sDx ;Amount of horizontal scroll to the right (in device units)
PUSH WORD sDy ;Amount of vertical scroll upward (in device units)
PUSH@ OTHER rc1Scroll ;Scroll rectangle
PUSH@ OTHER rc1Clip ;Clip rectangle
PUSH DWORD hrgnUpdateRgn ;Update region
PUSH@ OTHER rc1Update ;Update rectangle
PUSH WORD fsOptions ;Scroll options
CALL WinScrollWindow

Returns WORD sComplexity ;Complexity of resulting region/error indicator
```

WinSendDlgItemMsg

```
EXTRN WinSendDlgItemMsg:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndDlg ;Parent-window handle
PUSH WORD idItem ;Identity of the child window
PUSH WORD usMsgid ;Message identity
PUSH DWORD mpParam1 ;Message parameter 1
PUSH DWORD mpParam2 ;Message parameter 2
CALL WinSendDlgItemMsg

Returns DWORD mresReply ;Message-return data
```

WinSendMessage

```
EXTRN WinSendMessage:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD   hwnd      ;Window handle
PUSH WORD    usMsgid    ;Message identity
PUSH DWORD   mpParam1   ;Parameter 1
PUSH DWORD   mpParam2   ;Parameter 2
CALL WinSendMessage

Returns DWORD mresReply ;Message-return data
```

WinSetAccelerTable

```
EXTRN WinSetAccelerTable:FAR
INCL_WINACCELERATORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD   hab        ;Anchor-block handle
PUSH DWORD   haccelAcce ;Accelerator-table handle
PUSH DWORD   hwndFrame  ;Frame-window handle
CALL WinSetAccelerTable

Returns WORD  fSuccess   ;Success indicator
```

WinSetActiveWindow

```
EXTRN WinSetActiveWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD   hwndDesktop ;Desktop-window handle
PUSH DWORD   hwnd        ;Window handle
CALL WinSetActiveWindow

Returns WORD  fSuccess   ;Active-window-set indicator
```

WinSetCapture

```
EXTRN WinSetCapture:FAR
INCL_WININPUT equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD   hwndDesktop ;Desktop-window handle, or HWND_DESKTOP
PUSH DWORD   hwnd        ;Handle of the window that is to receive all pointing device mes-
                          ;sages
CALL WinSetCapture

Returns WORD  fSuccess   ;Success indicator
```

WinSetClassMsgInterest

```
EXTRN WinSetClassMsgInterest:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD   hab        ;Anchor-block handle
PUSH@ ASCIIZ  szClassName ;Window-class name
PUSH WORD    usMsgClass  ;Message class to have interest level set
PUSH WORD    sControl    ;Interest identifier for the message class
CALL WinSetClassMsgInterest

Returns WORD  fSuccess   ;Interest-changed indicator
```

WinSetClipbrdData

```
EXTRN WinSetClipbrdData:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD ulh ;Handle
PUSH WORD usfmt ;Format
PUSH WORD fsFmtInfo ;Information
CALL WinSetClipbrdData

Returns WORD fDataPlaced ;Data-placed indicator
```

WinSetClipbrdOwner

```
EXTRN WinSetClipbrdOwner:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hwnd ;Window handle of the new clipboard owner
CALL WinSetClipbrdOwner

Returns WORD fSuccess ;Success indicator
```

WinSetClipbrdViewer

```
EXTRN WinSetClipbrdViewer:FAR
INCL_WINCLIPBOARD equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hwndNewClipViewer ;Window handle of the new clipboard viewer
CALL WinSetClipbrdViewer

Returns WORD fSuccess ;Success indicator
```

WinSetCp

```
EXTRN WinSetCp:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hmq ;Message-queue handle
PUSH WORD usCodePage ;Code page
CALL WinSetCp

Returns WORD fSuccess ;Success indicator
```

WinSetDlgItemShort

```
EXTRN WinSetDlgItemShort:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwndDlg ;Parent-window handle
PUSH WORD idItem ;Identity of the child window whose text is to be changed
PUSH WORD usValue ;Integer value used to generate the dialog item text
PUSH WORD fSigned ;Sign indicator
CALL WinSetDlgItemShort

Returns WORD fSuccess ;Success indicator
```

WinSetDlgItemText

```
EXTRN WinSetDlgItemText:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwndDlg ;Parent-window handle
PUSH WORD idItem ;Identity of the child window whose text is to be set
PUSH@ ASCIIZ szText ;Source string
CALL WinSetDlgItemText

Returns WORD fSuccess ;Success indicator
```

WinSetErrorInfo

```
EXTRN WinSetErrorInfo:FAR

PUSH  DWORD  errIdError  ;Error identity to be set
PUSH  WORD   fsOptions   ;Options
PUSH  WORD   usargs      ;Parameter words
CALL  WinSetErrorInfo
```

WinSetFocus

```
EXTRN WinSetFocus:FAR
INCL_WININPUT equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hwndDesktop ;Desktop-window handle
PUSH  DWORD  hwndNewFocus ;Window handle to receive the focus
CALL  WinSetFocus

Returns WORD  fSuccess    ;Success indicator
```

WinSetHook

```
EXTRN WinSetHook:FAR
INCL_WINHOOKS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH  DWORD  hmq          ;Queue identity
PUSH  WORD   sHookType    ;Hook chain type
PUSH@ OTHER  HookProc     ;Address of the application hook procedure
PUSH  WORD   Module       ;Resource identity
CALL  WinSetHook

Returns WORD  fSuccess    ;Success indicator
```

WinSetKeyboardStateTable

```
EXTRN WinSetKeyboardStateTable:FAR
INCL_WININPUT equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDesktop ;Desktop-window handle
PUSH@ OTHER  abKeyStateTable ;Key state table
PUSH  WORD   fSet        ;Set indicator
CALL  WinSetKeyboardStateTable

Returns WORD  fSuccess    ;Success indicator
```

WinSetMsgInterest

```
EXTRN WinSetMsgInterest:FAR
INCL_WINMESSAGEGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd        ;Window handle
PUSH  WORD   usMsgClass   ;Message class to have interest level set
PUSH  WORD   sControl     ;Interest-identifier for the message class
CALL  WinSetMsgInterest

Returns WORD  fSuccess    ;Interest-changed indicator
```

WinSetMsgMode

```
EXTRN WinSetMsgMode:FAR
INCL_WINMESSAGEGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hab          ;Anchor block handle
PUSH@ ASCIIZ  szClassName ;Window class name
PUSH  WORD   sControl     ;Message mode identifier
CALL  WinSetMsgMode

Returns WORD  fSuccess    ;Message delay indicator
```


WinSetMultWindowPos

```
EXTRN WinSetMultWindowPos:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD    hab      ;Anchor-block handle
PUSH@  OTHER    aSwp     ;Array
PUSH  WORD     cCount   ;Window count
CALL  WinSetMultWindowPos

Returns WORD    fSuccess ;Positioning success indicator
```

WinSetOwner

```
EXTRN WinSetOwner:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD    hwnd      ;Window handle whose owner window is to be changed
PUSH  DWORD    hwndNewOwner ;Handle of the new owner
CALL  WinSetOwner

Returns WORD    fSuccess ;Success indicator
```

WinSetParent

```
EXTRN WinSetParent:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD    hwnd      ;Window handle
PUSH  DWORD    hwndNewParent ;New parent window handle
PUSH  WORD     fRedraw    ;Redraw indicator
CALL  WinSetParent

Returns WORD    fSuccess ;Parent-changed indicator
```

WinSetPointer

```
EXTRN WinSetPointer:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD    hwndDeskTop ;Desktop-window handle
PUSH  DWORD    hptrNewPointer ;New pointer handle
CALL  WinSetPointer

Returns WORD    fSuccess ;Pointer-updated indicator
```

WinSetPointerPos

```
EXTRN WinSetPointerPos:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD    hwndDeskTop ;Desktop-window handle
PUSH  WORD     sx          ;x position of pointer in screen coordinates
PUSH  WORD     sy          ;y position of pointer in screen coordinates
CALL  WinSetPointerPos

Returns WORD    fSuccess ;Pointer position updated indicator
```

WinSetPresParam

```
EXTRN WinSetPresParam:FAR
INCL_WINSYS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD    hwnd      ;Window handle
PUSH  DWORD    idAttrType ;Attribute type identity
PUSH  DWORD    cbAttrValueLen ;Byte count of the data passed in the AttrValue parameter
PUSH@  OTHER    AttrValue ;Attribute value
CALL  WinSetPresParam

Returns WORD    fSuccess ;Success indicator
```

WinSetRect

```
EXTRN WinSetRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  rcRect   ;Rectangle to be updated
PUSH  WORD   sLeft    ;Left edge of rectangle
PUSH  WORD   sBottom  ;Bottom edge of rectangle
PUSH  WORD   sRight   ;Right edge of rectangle
PUSH  WORD   sTop     ;Top edge of rectangle
CALL WinSetRect

Returns WORD  fSuccess ;Success indicator
```

WinSetRectEmpty

```
EXTRN WinSetRectEmpty:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  rcRect   ;Rectangle to be set empty
CALL WinSetRectEmpty

Returns WORD  fSuccess ;Success indicator
```

WinSetSynchroMode

```
EXTRN WinSetSynchroMode:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   sMode    ;Synchronization mode
CALL WinSetSynchroMode

Returns WORD  fSuccess ;Success indicator
```

WinSetSysColors

```
EXTRN WinSetSysColors:FAR
INCL_WINSYS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDesktop ;Desktop-window handle
PUSH  DWORD  flOptions    ;Options
PUSH  DWORD  ulFormat     ;Format of entries in the table, as follows
PUSH  DWORD  lStart       ;Starting system color index
PUSH  DWORD  ulTablen     ;Number of elements
PUSH@ OTHER  alTable      ;Table
CALL WinSetSysColors

Returns WORD  fSuccess ;Success indicator
```

WinSetSysModalWindow

```
EXTRN WinSetSysModalWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwndDesktop ;Desktop-window handle, or HWND_DESKTOP
PUSH  DWORD  hwnd        ;Handle of window to become system-modal window
CALL WinSetSysModalWindow

Returns WORD  fSuccess ;Success indicator
```

WinSetSysValue

```
EXTRN WinSetSysValue:FAR
INCL_WINSYS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndDesktop ;Desktop-window handle
PUSH WORD sValueid ;System-value identity
PUSH DWORD lValue ;System value
CALL WinSetSysValue

Returns WORD fSuccess ;Value-set indicator
```

WinSetWindowBits

```
EXTRN WinSetWindowBits:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH WORD sb ;Zero-based index of the value to be set
PUSH DWORD flData ;Bit data to store in the window words
PUSH DWORD flMask ;Bits to be written indicator
CALL WinSetWindowBits

Returns WORD fSuccess ;Success indicator
```

WinSetWindowPos

```
EXTRN WinSetWindowPos:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH DWORD hwndBehind ;Relative window-placement order
PUSH WORD sx ;Window position, x coordinate
PUSH WORD sy ;Window position, y coordinate
PUSH WORD scx ;Window size
PUSH WORD scy ;Window size
PUSH WORD fsOptions ;Window-positioning options
CALL WinSetWindowPos

Returns WORD fSuccess ;Repositioning indicator
```

WinSetWindowPtr

```
EXTRN WinSetWindowPtr:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH WORD sb ;Zero-based index into the window words
PUSH@ OTHER p ;Pointer value to store in the window words
CALL WinSetWindowPtr

Returns WORD fSuccess ;Success indicator
```

WinSetWindowText

```
EXTRN WinSetWindowText:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH@ ASCII szString ;Window text
CALL WinSetWindowText

Returns WORD fResult ;Text-updated indicator
```

WinSetWindowULong

```
EXTRN WinSetWindowULong:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH WORD sb ;Zero-based index of the value to be set
PUSH DWORD ulData ;Unsigned, long integer value to store in the window words
CALL WinSetWindowULong

Returns WORD fSuccess ;Success indicator
```

WinSetWindowUShort

```
EXTRN WinSetWindowUShort:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH WORD sb ;Zero-based index of the value to be set
PUSH WORD usData ;Unsigned, short integer value to store in the window words
CALL WinSetWindowUShort

Returns WORD fSuccess ;Success indicator
```

WinShowCursor

```
EXTRN WinShowCursor:FAR
INCL_WINCURSORS equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Handle of window to which the cursor belongs
PUSH WORD fShow ;Show indicator
CALL WinShowCursor

Returns WORD fSuccess ;Success indicator
```

WinShowPointer

```
EXTRN WinShowPointer:FAR
INCL_WINPOINTERS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndDesktop ;Desktop-window handle
PUSH WORD fShow ;Level-update indicator
CALL WinShowPointer

Returns WORD fSuccess ;Display-level-updated indicator
```

WinShowTrackRect

```
EXTRN WinShowTrackRect:FAR
INCL_WINTRACKRECT equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
PUSH WORD fShow ;Show indicator
CALL WinShowTrackRect

Returns WORD fSuccess ;Success indicator
```

WinShowWindow

```
EXTRN WinShowWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hwnd ;Window handle
PUSH WORD fNewVisibility ;New visibility state
CALL WinShowWindow

Returns WORD fSuccess ;Visibility changed indicator
```

WinStartTimer

```
EXTRN WinStartTimer:FAR
INCL_WINTIMER equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hwnd ;Window handle that is part of the timer identification
PUSH WORD idTimer ;Timer identifier
PUSH WORD usTimeout ;Delay time in milliseconds
CALL WinStartTimer

Returns WORD usRet ;Return code
```

WinStopTimer

```
EXTRN WinStopTimer:FAR
INCL_WINTIMER equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hwnd ;Window handle
PUSH WORD usTimer ;Timer identifier
CALL WinStopTimer

Returns WORD fSuccess ;Success indicator
```

WinSubclassWindow

```
EXTRN WinSubclassWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Handle of window that is being subclassed
PUSH@ OTHER NewWindowProc ;New window procedure
CALL WinSubclassWindow

Returns @OTHER OldWindowProc ;Old window procedure
```

WinSubstituteStrings

```
EXTRN WinSubstituteStrings:FAR
INCL_WINDIALOGS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Handle of window that processes the call
PUSH@ ASCIIZ szSrc ;Source string
PUSH WORD sDestMax ;Maximum number of characters returnable
PUSH@ ASCIIZ szDest ;Resultant string
CALL WinSubstituteStrings

Returns WORD sDestRet ;Actual number of characters returned
```

WinSubtractRect

```
EXTRN WinSubtractRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER rc1Dest ;Result
PUSH@ OTHER rc1Src1 ;First source rectangle
PUSH@ OTHER rc1Src2 ;Second source rectangle
CALL WinSubtractRect

Returns WORD fNonempty ;Not-empty indicator
```

WinTerminate

```
EXTRN WinTerminate:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM. Also in COMMON section

PUSH DWORD hab ;Anchor-block handle
CALL WinTerminate

Returns WORD fTerminated ;Termination indicator
```

WinTerminateApp

```
EXTRN WinTerminateApp:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD happ ;Application handle
CALL WinTerminateApp

Returns WORD fSuccess ;Success indicator
```

WinThrow

```
EXTRN WinThrow:FAR
INCL_WINCATCHTHROW equ 1 ;Or use INCL_WIN or INCL_PM

PUSH@ OTHER ctchbfCatchBuf ;Saved execution-environment buffer
PUSH WORD sThrowBack ;Return value
CALL WinThrow
```

WinTrackRect

```
EXTRN WinTrackRect:FAR
INCL_WINTRACKRECT equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle where tracking is to take place
PUSH DWORD hps ;Presentation-space handle
PUSH@ OTHER tiTrackinfo ;Track information
CALL WinTrackRect

Returns WORD fSuccess ;Success indicator
```

WinTranslateAccel

```
EXTRN WinTranslateAccel:FAR
INCL_WINACCELERATORS equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH DWORD hwnd ;Destination window
PUSH DWORD haccelAccel ;Accelerator-table handle
PUSH@ OTHER Qmsg ;Message to be translated
CALL WinTranslateAccel

Returns WORD fSuccess ;Success indicator
```

WinUnionRect

```
EXTRN WinUnionRect:FAR
INCL_WINRECTANGLES equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ OTHER rcIDest ;Bounding rectangle
PUSH@ OTHER rcISrc1 ;First source rectangle
PUSH@ OTHER rcISrc2 ;Second source rectangle
CALL WinUnionRect

Returns WORD fNonempty ;Non-empty indicator
```

WinUpdateWindow

```
EXTRN WinUpdateWindow:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwnd ;Window handle
CALL WinUpdateWindow

Returns WORD fSuccess ;Window-updated indicator
```

WinUpper

```
EXTRN WinUpper:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   usCodepage ;Code page
PUSH  WORD   usCountry ;Country code
PUSH@  ASCIIZ  szString ;String to be converted to uppercase
CALL  WinUpper

Returns WORD   usRetLen ;Length of converted string
```

WinUpperChar

```
EXTRN WinUpperChar:FAR
INCL_WINCOUNTRY equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   usCodepage ;Code page
PUSH  WORD   usCountry ;Country code
PUSH  WORD   usInchar  ;Character to be translated to uppercase
CALL  WinUpperChar

Returns WORD   usOutchar ;Translated character
```

WinValidateRect

```
EXTRN WinValidateRect:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd      ;Handle of window whose update region is changed
PUSH@  OTHER  rc1Rect   ;Rectangle to be subtracted from the window's update
                        ;region
PUSH  WORD   fIncludeClippedChildren ;Validation-scope indicator
CALL  WinValidateRect

Returns WORD   fSuccess ;Success indicator
```

WinValidateRegion

```
EXTRN WinValidateRegion:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hwnd      ;Handle of window whose update region is changed
PUSH  DWORD  hrgn      ;Handle of subtracted region
PUSH  WORD   fIncludeClippedChildren ;Validation-scope indicator
CALL  WinValidateRegion

Returns WORD   fSuccess ;Success indicator
```

WinWaitMsg

```
EXTRN WinWaitMsg:FAR
INCL_WINMESSAGEMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   usFirst   ;First message identity
PUSH  WORD   usLast    ;Last message identity
CALL  WinWaitMsg

Returns WORD   fSuccess ;Success indicator
```

WinWindowFromDC

```
EXTRN WinWindowFromDC:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hdc ;Device-context handle
CALL WinWindowFromDC

Returns DWORD hwnd ;Window handle
```

WinWindowFromID

```
EXTRN WinWindowFromID:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndParent ;Parent-window handle
PUSH WORD usIdentifier ;Identity of the child window
CALL WinWindowFromID

Returns DWORD hwnd ;Window handle
```

WinWindowFromPoint

```
EXTRN WinWindowFromPoint:FAR
INCL_WINWINDOWMGR equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hwndParent ;Window handle whose child windows are to be tested
PUSH@ OTHER ptlPoint ;The point to be tested
PUSH WORD fEnumChildren ;Test control
PUSH WORD fLock ;Lock control
CALL WinWindowFromPoint

Returns DWORD hwndFound ;Window handle beneath Point
```

WinWriteProfileData

```
EXTRN WinWriteProfileData:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor-block handle
PUSH@ ASCIIZ szAppName ;Text string
PUSH@ ASCIIZ szKeyName ;Text string
PUSH@ OTHER Value ;Value data
PUSH WORD usSize ;Size of value data
CALL WinWriteProfileData

Returns WORD fSuccess ;Success indicator
```

WinWriteProfileString

```
EXTRN WinWriteProfileString:FAR
INCL_WINSHELLDATA equ 1 ;Or use INCL_WIN or INCL_PM

PUSH DWORD hab ;Anchor block handle
PUSH@ ASCIIZ szAppName ;Text string
PUSH@ ASCIIZ szKeyName ;Text string
PUSH@ ASCIIZ szValue ;Text string
CALL WinWriteProfileString

Returns WORD fSuccess ;Success indicator
```

Chapter 10. Functions Supplied by Applications

DialogProc

```
EXTRN DialogProc:FAR

PUSH  DWORD  hwnd      ;Handle of the window to which the message applies
PUSH  WORD   usmsg     ;Message identity
PUSH  DWORD  mpParam1  ;Message parameter 1
PUSH  DWORD  mpParam2  ;Message parameter 2
CALL  DialogProc

Returns DWORD mresReply ;Message-return data
```

WndProc

```
EXTRN WndProc:FAR

PUSH  DWORD  hwnd      ;Window handle
PUSH  WORD   usmsg     ;Message identity
PUSH  DWORD  mpParam1  ;Message parameter 1
PUSH  DWORD  mpParam2  ;Message parameter 2
CALL  WndProc

Returns DWORD mresReply ;Message-return data
```

CodePageChangeHook

```
EXTRN CodePageChangeHook:FAR

PUSH  DWORD  hmq       ;Message-queue handle
PUSH  WORD   usOldCodePage ;Previous code page
PUSH  WORD   usNewCodePage ;New code page
CALL  CodePageChangeHook
```

HelpHook

```
EXTRN HelpHook:FAR

PUSH  DWORD  hab        ;Anchor-block handle
PUSH  WORD   sMode      ;Help mode
PUSH  WORD   sTopic     ;Topic identifier
PUSH  WORD   sSubTopic  ;Subtopic identifier
PUSH@  OTHER  rcIPosition ;Rectangle
CALL  HelpHook

Returns WORD  f          ;Indicator as to whether next hook in the chain is called
```

InputHook

```
EXTRN InputHook:FAR

PUSH  DWORD  hab        ;Anchor-block handle
PUSH@  OTHER  Qmsg      ;A QMSG data structure
PUSH  WORD   fsOptions  ;Message removal options
CALL  InputHook

Returns WORD  fProcessed ;Processed indicator
```

JournalPlaybackHook

EXTRN JournalPlaybackHook:FAR

```
PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  qmsg     ;Data structure where the message to be played back is returned
PUSH  WORD   fSkip    ;Indicator as to whether the next message should be played back
CALL  JournalPlaybackHook
```

Returns DWORD lTime ;Waiting time

JournalRecordHook

EXTRN JournalRecordHook:FAR

```
PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  qmsg     ;Data structure that contains the message to be recorded
CALL  JournalRecordHook
```

Returns WORD fSuccess ;The return value from this hook is ignored

LoaderHook

EXTRN LoaderHook:FAR

```
PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   sContext  ;Origin of call to hook
PUSH@ ASCIIZ  szlibname ;Library name
PUSH@ DWORD  hlibLibhandle ;Library handle
PUSH@ ASCIIZ  szprocname ;Procedure name
PUSH@ OTHER  wndproc   ;Window procedure identifier
PUSH@ WORD   fSuccess  ;Success indicator
CALL  LoaderHook
```

Returns WORD fProcessed ;Processing indicator

MsgCtlHook

EXTRN MsgCtlHook:FAR

```
PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   sContext  ;Origin of call to hook
PUSH  DWORD  hwnd      ;Window handle
PUSH@ ASCIIZ  szClassName ;Window class name
PUSH  WORD   usMsgClass ;Message class
PUSH  WORD   sControl  ;Control setting
PUSH@ WORD   fSuccess  ;Success indicator
CALL  MsgCtlHook
```

Returns WORD fProcessed ;Processing indicator

MsgFilterHook

EXTRN MsgFilterHook:FAR

```
PUSH  DWORD  hab      ;Anchor-block handle
PUSH@ OTHER  Qmsg     ;A queue message data structure
PUSH  WORD   usContext ;Context in which the hook has been called
CALL  MsgFilterHook
```

Returns WORD fProcessed ;Processed indicator

RegisterUserMsg

EXTRN RegisterUserMsg:FAR

```
PUSH  DWORD  hab      ;Anchor-block handle
PUSH  WORD   sContext ;Origin of call to hook
PUSH  WORD   usMsgid  ;Message identifier
PUSH  WORD   sType1   ;Data type of message-parameter 1
PUSH  WORD   sDir1    ;Direction of message-parameter 1
PUSH  WORD   sType2   ;Data type of message-parameter 2
PUSH  WORD   sDir2    ;Direction of message-parameter 2
PUSH  WORD   sTyper   ;Data type of message reply
PUSH  WORD   sCount   ;Number of elements
PUSH@  OTHER asTypes  ;Data types of structure components
PUSH@  WORD   fSuccess ;Success indicator
CALL  RegisterUserMsg
```

Returns WORD fProcessed ;Processing indicator

SendMsgHook

EXTRN SendMsgHook:FAR

```
PUSH  DWORD  hab      ;Anchor-block handle
PUSH@  OTHER smhssmh  ;Send message hook structure
PUSH  WORD   fInterTask ;Intertask indicator
CALL  SendMsgHook
```


Index

A

ACCEL 2-1
ACCELTABLE 2-1
addressing elements in arrays 1-3
ARCPARAM 2-1
ARCPARAMS 2-1
AREABUNDLE 2-1
arrays
 addressing elements in 1-3
ASCIIZ data type 1-3
ATOM 2-1

B

BANDRECT 2-1
BITMAPINFO 2-2
BITMAPINFOHEADER 2-2
BIT16 2-2
BIT32 2-2
BIT8 2-2
BOOL 2-2
BTNCDATA 2-2
BUFFER 2-2
BUNDLE 2-3
BYTE 2-3

C

call characteristics 1-2
call format 1-2
CALL operation 1-2
CATCHBUF 2-3
CHAR 2-3
CHARBUNDLE 2-3
CLASSINFO 2-3
CodePageChangeHook 10-1
COUNT2 2-3
COUNT2B 2-3
COUNT2CH 2-3
COUNT4 2-3
COUNT4B 2-3
CPID 2-3
CREATEPARAMS 2-3
CREATESTRECT 2-4
CTLDATA 2-4
CURSORINFO 2-4

D

DDEINIT 2-4
DDESTRUCT 2-4
DevCloseDC 3-1
DevEscape 3-1
DEVOPENDATA 2-4
DevOpenDC 3-1
DEVOPENSTRUC 2-5
DevPostDeviceModes 3-1
DevQueryCaps 3-2
DevQueryDeviceNames 3-2
DevQueryHardcopyCaps 3-2
DialogProc 10-1

DLGTEMPLATE 2-5
DLGTITEM 2-5
DRIVDATA 2-5
DWORD data type 1-3
dynamic linking 1-1

E

ENTRYFDATA 2-6
ERRINFO 2-6
ERRORID 2-6

F

FATTRS 2-6
FFDESCS 2-6
FIXED 2-6
FONTMETRICS 2-7
FRAMECDATA 2-8
function call rules 1-2
function calls 1-2

G

GpiAssociate 4-1
GpiBeginArea 4-1
GpiBeginElement 4-1
GpiBeginPath 4-1
GpiBitBit 4-1
GpiBox 4-2
GpiCallSegmentMatrix 4-2
GpiCharString 4-2
GpiCharStringAt 4-2
GpiCharStringPos 4-2
GpiCharStringPosAt 4-3
GpiCloseFigure 4-3
GpiCloseSegment 4-3
GpiCombineRegion 4-3
GpiComment 4-3
GpiConvert 4-4
GpiCopyMetaFile 4-4
GpiCorrelateChain 4-4
GpiCorrelateFrom 4-4
GpiCorrelateSegment 4-5
GpiCreateBitmap 4-5
GpiCreateLogColorTable 4-5
GpiCreateLogFont 4-5
GpiCreatePS 4-6
GpiCreateRegion 4-6
GpiDeleteBitmap 4-6
GpiDeleteElement 4-6
GpiDeleteElementRange 4-6
GpiDeleteElementsBetweenLabels 4-6
GpiDeleteMetaFile 4-7
GpiDeleteSegment 4-7
GpiDeleteSegments 4-7
GpiDeleteSetId 4-7
GpiDestroyPS 4-7
GpiDestroyRegion 4-7
GpiDrawChain 4-8
GpiDrawDynamics 4-8
GpiDrawFrom 4-8

GpiDrawSegment	4-8	GpiQueryDefaultViewMatrix	4-20
GpiElement	4-8	GpiQueryDefCharBox	4-20
GpiEndArea	4-8	GpiQueryDefTag	4-20
GpiEndElement	4-9	GpiQueryDefViewingLimits	4-20
GpiEndPath	4-9	GpiQueryDevice	4-20
GpiEqualRegion	4-9	GpiQueryDeviceBitmapFormats	4-20
GpiErase	4-9	GpiQueryDrawControl	4-21
GpiErrorSegmentData	4-9	GpiQueryDrawingMode	4-21
GpiExcludeClipRectangle	4-9	GpiQueryEditMode	4-21
GpiFillPath	4-10	GpiQueryElement	4-21
GpiFullArc	4-10	GpiQueryElementPointer	4-21
GpiGetData	4-10	GpiQueryElementType	4-21
GpiImage	4-10	GpiQueryFontFileDescriptions	4-22
GpiIntersectClipRectangle	4-10	GpiQueryFontMetrics	4-22
GpiLabel	4-11	GpiQueryFonts	4-22
GpiLine	4-11	GpiQueryGraphicsField	4-22
GpiLoadBitmap	4-11	GpiQueryInitialSegmentAttrs	4-22
GpiLoadFonts	4-11	GpiQueryKerningPairs	4-23
GpiLoadMetaFile	4-11	GpiQueryLineEnd	4-23
GpiMarker	4-11	GpiQueryLineJoin	4-23
GpiModifyPath	4-12	GpiQueryLineType	4-23
GpiMove	4-12	GpiQueryLineWidth	4-23
GpiOffsetClipRegion	4-12	GpiQueryLineWidthGeom	4-23
GpiOffsetElementPointer	4-12	GpiQueryLogColorTable	4-24
GpiOffsetRegion	4-12	GpiQueryMarker	4-24
GpiOpenSegment	4-12	GpiQueryMarkerBox	4-24
GpiOutlinePath	4-13	GpiQueryMarkerSet	4-24
GpiPaintRegion	4-13	GpiQueryMetaFileBits	4-24
GpiPartialArc	4-13	GpiQueryMetaFileLength	4-24
GpiPlayMetaFile	4-13	GpiQueryMix	4-25
GpiPointArc	4-13	GpiQueryModelTransformMatrix	4-25
GpiPolyFillet	4-14	GpiQueryNearestColor	4-25
GpiPolyFilletSharp	4-14	GpiQueryNumberSetIds	4-25
GpiPolyLine	4-14	GpiQueryPageViewport	4-25
GpiPolyMarker	4-14	GpiQueryPattern	4-25
GpiPolySpline	4-14	GpiQueryPatternRefPoint	4-26
GpiPop	4-14	GpiQueryPatternSet	4-26
GpiPtInRegion	4-15	GpiQueryPel	4-26
GpiPtVisible	4-15	GpiQueryPickAperturePosition	4-26
GpiPutData	4-15	GpiQueryPickApertureSize	4-26
GpiQueryArcParams	4-15	GpiQueryPS	4-26
GpiQueryAttrMode	4-15	GpiQueryRealColors	4-27
GpiQueryAttrs	4-15	GpiQueryRegionBox	4-27
GpiQueryBackColor	4-16	GpiQueryRegionRects	4-27
GpiQueryBackMix	4-16	GpiQueryRGBColor	4-27
GpiQueryBitmapBits	4-16	GpiQuerySegmentAttrs	4-27
GpiQueryBitmapDimension	4-16	GpiQuerySegmentNames	4-28
GpiQueryBitmapHandle	4-16	GpiQuerySegmentPriority	4-28
GpiQueryBitmapParameters	4-16	GpiQuerySegmentTransformMatrix	4-28
GpiQueryBoundaryData	4-17	GpiQuerySetIds	4-28
GpiQueryCharAngle	4-17	GpiQueryStopDraw	4-28
GpiQueryCharBox	4-17	GpiQueryTag	4-29
GpiQueryCharDirection	4-17	GpiQueryTextBox	4-29
GpiQueryCharMode	4-17	GpiQueryViewingLimits	4-29
GpiQueryCharSet	4-17	GpiQueryViewingTransformMatrix	4-29
GpiQueryCharShear	4-17	GpiQueryWidthTable	4-29
GpiQueryCharStringPos	4-18	GpiRealizeColorTable	4-29
GpiQueryCharStringPosAt	4-18	GpiRectInRegion	4-30
GpiQueryClipBox	4-18	GpiRectVisible	4-30
GpiQueryClipRegion	4-18	GpiRemoveDynamics	4-30
GpiQueryColor	4-18	GpiResetBoundaryData	4-30
GpiQueryColorData	4-19	GpiResetPS	4-30
GpiQueryColorIndex	4-19	GpiRestorePS	4-30
GpiQueryCp	4-19	GpiRotate	4-31
GpiQueryCurrentPosition	4-19	GpiSaveMetaFile	4-31
GpiQueryDefArcParams	4-19	GpiSavePS	4-31
GpiQueryDefAttrs	4-19	GpiScale	4-31

- GpiSetArcParams 4-31
- GpiSetAttrMode 4-31
- GpiSetAttrs 4-32
- GpiSetBackColor 4-32
- GpiSetBackMix 4-32
- GpiSetBitmap 4-32
- GpiSetBitmapBits 4-32
- GpiSetBitmapDimension 4-32
- GpiSetBitmapId 4-33
- GpiSetCharAngle 4-33
- GpiSetCharBox 4-33
- GpiSetCharDirection 4-33
- GpiSetCharMode 4-33
- GpiSetCharSet 4-33
- GpiSetCharShear 4-34
- GpiSetClipPath 4-34
- GpiSetClipRegion 4-34
- GpiSetColor 4-34
- GpiSetCp 4-34
- GpiSetCurrentPosition 4-34
- GpiSetDefArcParams 4-35
- GpiSetDefAttrs 4-35
- GpiSetDefaultViewMatrix 4-35
- GpiSetDefTag 4-35
- GpiSetDefViewingLimits 4-35
- GpiSetDrawControl 4-35
- GpiSetDrawingMode 4-36
- GpiSetEditMode 4-36
- GpiSetElementPointer 4-36
- GpiSetElementPointerAtLabel 4-36
- GpiSetGraphicsField 4-36
- GpiSetInitialSegmentAttrs 4-36
- GpiSetLineEnd 4-37
- GpiSetLineJoin 4-37
- GpiSetLineType 4-37
- GpiSetLineWidth 4-37
- GpiSetLineWidthGeom 4-37
- GpiSetMarker 4-37
- GpiSetMarkerBox 4-38
- GpiSetMarkerSet 4-38
- GpiSetMetaFileBits 4-38
- GpiSetMix 4-38
- GpiSetModelTransformMatrix 4-38
- GpiSetPageViewport 4-38
- GpiSetPattern 4-39
- GpiSetPatternRefPoint 4-39
- GpiSetPatternSet 4-39
- GpiSetPel 4-39
- GpiSetPickAperturePosition 4-39
- GpiSetPickApertureSize 4-39
- GpiSetPS 4-40
- GpiSetRegion 4-40
- GpiSetSegmentAttrs 4-40
- GpiSetSegmentPriority 4-40
- GpiSetSegmentTransformMatrix 4-40
- GpiSetStopDraw 4-41
- GpiSetTag 4-41
- GpiSetViewingLimits 4-41
- GpiSetViewingTransformMatrix 4-41
- GpiStrokePath 4-41
- GpiTranslate 4-41
- GpiUnloadFonts 4-42
- GpiUnrealizeColorTable 4-42
- GpiWCBitBit 4-42
- GRADIENT 2-8
- GRADIENTL 2-8

H

- HAB 2-8
- HACCEL 2-8
- HANDLE 2-8
- HAPP 2-8
- HATOMTBL 2-8
- HBITMAP 2-8
- HCINFO 2-8
- HDC 2-8
- HelpHook 10-1
- HENUM 2-8
- HHEAP 2-8
- HINI 2-8
- HLIB 2-8
- HMF 2-8
- HMODULE 2-9
- HMQ 2-9
- HPOINTER 2-9
- HPROC 2-9
- HPROGARRAY 2-9
- HPROGRAM 2-9
- HPS 2-9
- HRGN 2-9
- HSEM 2-9
- HSPL 2-9
- HSWITCH 2-9
- HVPS 2-9
- HWND 2-9

I

- IDENTITY 2-9
- IDENTITY4 2-9
- IMAGEBUNDLE 2-9
- INDEX2 2-9
- InputHook 10-1
- IPT 2-9

J

- JournalPlaybackHook 10-2
- JournalRecordHook 10-2

K

- kerning 2-9
- KERNINGPAIRS 2-9

L

- LENGTH2 2-9
- LENGTH4 2-10
- LHANDLE 2-10
- LINEBUNDLE 2-10
- LoaderHook 10-2
- LONG 2-10

M

- mapping metalanguage data types to MASM 1-3
- MARGSTRUCT 2-10
- MARKERBUNDLE 2-10
- MASM bindings
 - header files 1-1
- MASM data types

MASM data types (*continued*)
 mapping from metalanguage data types 1-3
 MATRIX 2-10
 MATRIXLF 2-11
 MENUITEM 2-11
 metalanguage data types
 mapping to MASM data types 1-3
 MLECTLDATA 2-11
 MLE_SEARCHDATA 2-11
 MPARAM 2-12
 MQINFO 2-12
 MRESULT 2-12
 MsgCtlHook 10-2
 MsgFilterHook 10-2
 MT 2-12
 MTI 2-12

N

NPBYTE 2-12

O

OFFSET2B 2-12
 OVERFLOW 2-12
 OWNERITEM 2-13

P

PARAM 2-13
 PBYTE 2-13
 PCH 2-13
 PDEVOPENDATA 2-13
 PFN 2-13
 PFNWP 2-13
 PIBSTRUCT 2-13
 Picchg 5-1
 PicPrint 5-1
 PID 2-14
 PIX 2-14
 POINT 2-14
 POINTERINFO 2-14
 POINTL 2-14
 POINTS 2-14
 PQMOPENDATA 2-14
 PRESDATA 2-14
 PRESPARAMS 2-14
 PrfAddProgram 6-1
 PrfChangeProgram 6-1
 PrfCloseProfile 6-1
 PrfCreateGroup 6-1
 PrfDestroyGroup 6-1
 PrfOpenProfile 6-2
 PRFFPROFILE 2-14
 PrfQueryDefinition 6-2
 PrfQueryProfile 6-2
 PrfQueryProfileData 6-2
 PrfQueryProfileInt 6-2
 PrfQueryProfileSize 6-3
 PrfQueryProfileString 6-3
 PrfQueryProgramCategory 6-3
 PrfQueryProgramHandle 6-3
 PrfQueryProgramTitles 6-3
 PrfRemoveProgram 6-4
 PrfReset 6-4
 PrfWriteProfileData 6-4

PrfWriteProfileString 6-4
 PROC 2-14
 PROGCATEGORY 2-14
 PROGDETAILS 2-15
 PROGRAMENTRY 2-15
 PROGTITLE 2-15
 PROGTYPE 2-15
 PROPERTY2 2-15
 PROPERTY4 2-15
 PSZ 2-15
 PUSH operation 1-2
 PUSH@ operation 1-2
 PVOID 2-15

Q

QMOPENDATA 2-15
 QMSG 2-16

R

RECT 2-16
 RECTL 2-16
 RegisterUserMsg 10-3
 RESID 2-16
 RGB 2-16
 RGNRECT 2-16
 ROF 2-16
 ROL 2-16

S

SBCDATA 2-17
 SEGOFF 2-17
 SendMsgHook 10-3
 SHORT 2-17
 SIZEF 2-17
 SIZEL 2-17
 SIZEROF 2-17
 SIZEROL 2-17
 SMHSTRUCT 2-17
 SplQmAbort 7-1
 SplQmClose 7-1
 SplQmEndDoc 7-1
 SplQmOpen 7-1
 SplQmStartDoc 7-1
 SplQmWrite 7-1
 SplQpInstall 7-2
 SplQpQueryDt 7-2
 standard data types 1-3
 STORAGE 2-17
 STR 2-17
 STRCOND 2-17
 STRL 2-18
 STRLLIST 2-18
 STR16 2-18
 STR32 2-18
 STR64 2-18
 STR8 2-18
 SWBLOCK 2-18
 SWCNTRL 2-18
 SWENTRY 2-18
 SWP 2-18
 SZ 2-18

T

TID 2-18
TIME 2-19
TRACKINFO 2-19

U

UCHAR 2-19
ULONG 2-19
USERBUTTON 2-19
USHORT 2-19

V

VioAssociate 8-1
VioCreateLogFont 8-1
VioCreatePS 8-1
VioDeleteSetId 8-1
VioDestroyPS 8-1
VioGetDeviceCellSize 8-2
VioGetOrg 8-2
VioQueryFonts 8-2
VioQuerySetIds 8-2
VioSetDeviceCellSize 8-2
VioSetOrg 8-3
VioShowPS 8-3
VOID 2-19

W

WIDTH4 2-19
WinAddAtom 9-1
WinAddProgram 9-1
WinAddSwitchEntry 9-1
WinAlarm 9-1
WinAllocMem 9-1
WinAvailMem 9-2
WinBeginEnumWindows 9-2
WinBeginPaint 9-2
WinBroadcastMsg 9-2
WinCalcFrameRect 9-2
WinCallMsgFilter 9-2
WinCancelShutdown 9-3
WinCatch 9-3
WinChangeSwitchEntry 9-3
WinCloseClipbrd 9-3
WinCompareStrings 9-3
WinCopyAccelTable 9-3
WinCopyRect 9-4
WinCpTranslateChar 9-4
WinCpTranslateString 9-4
WinCreateAccelTable 9-4
WinCreateAtomTable 9-4
WinCreateCursor 9-5
WinCreateDlg 9-5
WinCreateFrameControls 9-5
WinCreateGroup 9-5
WinCreateHeap 9-6
WinCreateMenu 9-6
WinCreateMsgQueue 9-6
WinCreatePointer 9-6
WinCreatePointerIndirect 9-6
WinCreateStdWindow 9-7
WinCreateSwitchEntry 9-7
WinCreateWindow 9-7

WinDdeInitiate 9-7
WinDdePostMsg 9-8
WinDdeRespond 9-8
WinDefAVioWindowProc 9-8
WinDefDlgProc 9-8
WinDefWindowProc 9-8
WinDeleteAtom 9-9
WinDeleteLibrary 9-9
WinDeleteProcedure 9-9
WinDestroyAccelTable 9-9
WinDestroyAtomTable 9-9
WinDestroyCursor 9-9
WinDestroyHeap 9-9
WinDestroyMsgQueue 9-10
WinDestroyPointer 9-10
WinDestroyWindow 9-10
WinDismissDlg 9-10
WinDispatchMsg 9-10
WinDlgBox 9-10
WinDrawBitmap 9-11
WinDrawBorder 9-11
WinDrawPointer 9-11
WinDrawText 9-11
WinEmptyClipbrd 9-12
WinEnablePhysInput 9-12
WinEnableWindow 9-12
WinEnableWindowUpdate 9-12
WinEndEnumWindows 9-12
WinEndPaint 9-12
WinEnumClipbrdFmts 9-12
WinEnumDlgItem 9-13
WinEqualRect 9-13
WinExcludeUpdateRegion 9-13
WinFillRect 9-13
WinFindAtom 9-13
WinFlashWindow 9-13
WinFocusChange 9-14
WinFreeErrorInfo 9-14
WinFreeMem 9-14
WinGetClipPS 9-14
WinGetCurrentTime 9-14
WinGetDlgMsg 9-14
WinGetErrorInfo 9-15
WinGetKeyState 9-15
WinGetLastError 9-15
WinGetMinPosition 9-15
WinGetMsg 9-15
WinGetNextWindow 9-15
WinGetPhysKeyState 9-16
WinGetPS 9-16
WinGetScreenPS 9-16
WinGetSysBitmap 9-16
WinInflateRect 9-16
WinInitialize 9-16
WinInSendMessage 9-17
WinInstStartApp 9-17
WinIntersectRect 9-17
WinInvalidateRect 9-17
WinInvalidateRegion 9-17
WinInvertRect 9-18
WinIsChild 9-18
WinIsRectEmpty 9-18
WinIsThreadActive 9-18
WinIsWindow 9-18
WinIsWindowEnabled 9-18
WinIsWindowShowing 9-18
WinIsWindowVisible 9-19

WinLoadAccelTable	9-19	WinQuerySwitchList	9-31
WinLoadDlg	9-19	WinQuerySysColor	9-31
WinLoadLibrary	9-19	WinQuerySysModalWindow	9-31
WinLoadMenu	9-19	WinQuerySysPointer	9-31
WinLoadPointer	9-19	WinQuerySystemAtomTable	9-31
WinLoadProcedure	9-20	WinQuerySysValue	9-32
WinLoadString	9-20	WinQueryTaskSizePos	9-32
WinLockHeap	9-20	WinQueryTaskTitle	9-32
WinLockVisRegions	9-20	WinQueryUpdateRect	9-32
WinLockWindow	9-20	WinQueryUpdateRegion	9-32
WinLockWindowUpdate	9-20	WinQueryVersion	9-32
WinMakePoints	9-21	WinQueryWindow	9-33
WinMakeRect	9-21	WinQueryWindowDC	9-33
WinMapDlgPoints	9-21	WinQueryWindowLockCount	9-33
WinMapWindowPoints	9-21	WinQueryWindowPos	9-33
WinMessageBox	9-21	WinQueryWindowProcess	9-33
WinMsgMuxSemWait	9-22	WinQueryWindowPtr	9-33
WinMsgSemWait	9-22	WinQueryWindowRect	9-34
WinMultWindowFromIDs	9-22	WinQueryWindowText	9-34
WinNextChar	9-22	WinQueryWindowTextLength	9-34
WinOffsetRect	9-22	WinQueryWindowULong	9-34
WinOpenClipbrd	9-22	WinQueryWindowUShort	9-34
WinOpenWindowDC	9-23	WinReallocMem	9-34
WinPeekMsg	9-23	WinRegisterClass	9-35
WinPostMsg	9-23	WinRegisterUserDatatype	9-35
WinPostQueueMsg	9-23	WinRegisterUserMsg	9-35
WinPrevChar	9-23	WinRegisterWindowDestroy	9-35
WinProcessDlg	9-24	WinReleaseHook	9-35
WinPtlInRect	9-24	WinReleasePS	9-36
WinQueryAccelTable	9-24	WinRemovePresParam	9-36
WinQueryActiveWindow	9-24	WinRemoveSwitchEntry	9-36
WinQueryAnchorBlock	9-24	WinScrollWindow	9-36
WinQueryAtomLength	9-24	WinSendDlgItemMsg	9-36
WinQueryAtomName	9-25	WinSendMsg	9-37
WinQueryAtomUsage	9-25	WinSetAccelTable	9-37
WinQueryCapture	9-25	WinSetActiveWindow	9-37
WinQueryClassInfo	9-25	WinSetCapture	9-37
WinQueryClassName	9-25	WinSetClassMsgInterest	9-37
WinQueryClipbrdData	9-25	WinSetClipbrdData	9-38
WinQueryClipbrdFmtInfo	9-26	WinSetClipbrdOwner	9-38
WinQueryClipbrdOwner	9-26	WinSetClipbrdViewer	9-38
WinQueryClipbrdViewer	9-26	WinSetCp	9-38
WinQueryCp	9-26	WinSetDlgItemShort	9-38
WinQueryCpList	9-26	WinSetDlgItemText	9-38
WinQueryCursorInfo	9-26	WinSetErrorInfo	9-39
WinQueryDefinition	9-27	WinSetFocus	9-39
WinQueryDesktopWindow	9-27	WinSetHook	9-39
WinQueryDlgItemShort	9-27	WinSetKeyboardStateTable	9-39
WinQueryDlgItemText	9-27	WinSetMsgInterest	9-39
WinQueryDlgItemTextLength	9-27	WinSetMsgMode	9-39
WinQueryFocus	9-27	WinSetMultWindowPos	9-40
WinQueryMsgPos	9-28	WinSetOwner	9-40
WinQueryMsgTime	9-28	WinSetParent	9-40
WinQueryObjectWindow	9-28	WinSetPointer	9-40
WinQueryPointer	9-28	WinSetPointerPos	9-40
WinQueryPointerInfo	9-28	WinSetPresParam	9-40
WinQueryPointerPos	9-28	WinSetRect	9-41
WinQueryPresParam	9-29	WinSetRectEmpty	9-41
WinQueryProfileData	9-29	WinSetSynchroMode	9-41
WinQueryProfileInt	9-29	WinSetSysColors	9-41
WinQueryProfileSize	9-29	WinSetSysModalWindow	9-41
WinQueryProfileString	9-30	WinSetSysValue	9-42
WinQueryProgramTitles	9-30	WinSetWindowBits	9-42
WinQueryQueueInfo	9-30	WinSetWindowPos	9-42
WinQueryQueueStatus	9-30	WinSetWindowPtr	9-42
WinQuerySwitchEntry	9-30	WinSetWindowText	9-42
WinQuerySwitchHandle	9-31	WinSetWindowULong	9-43

- WinSetWindowUShort 9-43
- WinShowCursor 9-43
- WinShowPointer 9-43
- WinShowTrackRect 9-43
- WinShowWindow 9-43
- WinStartTimer 9-44
- WinStopTimer 9-44
- WinSubclassWindow 9-44
- WinSubstituteStrings 9-44
- WinSubtractRect 9-44
- WinTerminate 9-44
- WinTerminateApp 9-45
- WinThrow 9-45
- WinTrackRect 9-45
- WinTranslateAccel 9-45
- WinUnionRect 9-45
- WinUpdateWindow 9-45
- WinUpper 9-46
- WinUpperChar 9-46
- WinValidateRect 9-46
- WinValidateRegion 9-46
- WinWaitMsg 9-46
- WinWindowFromDC 9-47
- WinWindowFromID 9-47
- WinWindowFromPoint 9-47
- WinWriteProfileData 9-47
- WinWriteProfileString 9-47
- WNDPARAMS 2-19
- WNDPROC 2-19, 10-1
- WORD data type 1-3
- WPOINT 2-20
- WRECT 2-20

X

- XYWINSIZE 2-20

IBM United Kingdom
International Products Limited
PO Box 41, North Harbour
Portsmouth, PO6 3AU
England

Printed in Denmark by
visoprint as

IBM